CONTENT-AWARE APPROACHES FOR DIGITAL VIDEO ADAPTATION, SUMMARIZATION AND COMPRESSION

By

TAORAN LU

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2010

To everyone that motivated, helped and encouraged me on the way towards my PhD

ACKNOWLEDGMENTS

First and most, I express my sincere gratitude to my Ph.D. advisor Dr. Dapeng Oliver Wu for his support that made possible the completion of this work. Without his guidance, imagination, and enthusiasm, passion, which I admire, this dissertation would not have been possible. I also thank my committee member Dr. John Harris, Dr. Scott Banks and Dr. Yijun Sun for their valuable discussions and comments to improve the quality of this dissertation.

My deepest appreciation also goes to Dr.Thomas Deselaers and Philippe Dreuw from Rheinisch-Westfalische Technische Hochschule (RWTH) Aachen University, GE, for their kind help with my experiments and subjective evaluations on video adaptation. I also would like to thank the compression group in Technicolor Research and Innovation. I would like to thank Jill Boyce, Peng Yin, Xiaoan Lu, Qian Xu, Yunfei Zheng, Liwei Guo, Dong Tian and Polin Lai, for their support and guidance on my research on H.264 video codec and next generation video coding.

During my course on Ph.D. research, I interact with my colleagues and I benefit a lot from the valuable discussions on research and life at large. Especially, I thank former and current group members Dr. Xiaochen Li, Dr. Jun Xu, Dr. Bing Han, Dr. Xihua Dong, Zhifeng Chen, Zheng Yuan, Huanghuang Li, Qian Chen, Zongrui Ding, Lei Yang, and Yuejia He. Last but not least, I thank my parents for their love and support throughout my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

CONTENT-AWARE APPROACHES FOR DIGITAL VIDEO ADAPTATION,
SUMMARIZATION AND COMPRESSION

By

Taoran Lu

December 2010

Chair: Dapeng Oliver Wu
Major: Electrical and Computer Engineering

In this dissertation, we mainly present our work on three challenging problems of digital video applications: video compression, summarization and adaptation. Unlike conventional techniques, we focus on the video content modeling and investigate how the characteristics of human attention will help solving these problems. We denote these approaches “content-aware” approaches and present our innovations in the main body of this dissertation.

The first problem is content-aware video adaptation. We employ saliency analysis, which generates a saliency map to indicate the relative importance of pixels within a frame for human attention modeling. We also propose a nonlinear saliency map fusing approach that considers human perceptual characteristics. To effectively map the important content from source to the target display, we propose to have both intra-frame visual considerations and inter-frame visual considerations, where intra-considerations focus on measuring the information loss within a frame, and inter-considerations emphasis the visual smoothness between frames. The mapping problem is formulated as a shortest path problem and is solved with dynamic programming.

The second problem is content-aware video summarization. We introduce an automatic video summarization approach that includes unsupervised learning of original video-audio concept primitives and hierarchical (both frame and shot levels) skimming. For video concept mining, we propose a novel model using bag-of-word (BoW) shot

features. We further design a hierarchical video summarization framework which jointly considers content completeness, saliency, smoothing and scalability.

Another problem that we investigate is content-aware video compression. We study this problem in two aspects. In one aspect, we propose a content-aware framework and formulate the constraint optimization of rate-distortion as a resource allocation problem, where bit-allocation is adjusted differently at two levels: region-of-interest (ROI) and non-ROI, intra frames and inter frames. The results exhibit better visual quality as well as objective quality improvement. In the other aspect, we aim at improving the coding efficiency of existing H.264 intra prediction. We incorporate a reverse encoding order with geometric analysis for binary transition points on block boundaries to explicitly derive the prediction direction. Besides, we design and implement a video coding parameter analyzer to facilitate the development of new coding tools for state-of-the-art and next-generation video compression standards.

CHAPTER 1
INTRODUCTION

1.1 Motivation

This decade is the decade of multimedia. Thanks to the fast development of the Internet, the network volume has increased dramatically, and the number of supported users has also grown apace. In the consumer electronics industry, the advanced techniques in digital cameras and camcorders have made it easy and pleasant for everyone to capture and store a digital video. There is an obvious trend that more and more people start to watch and share digital videos online with their family, friends or even strangers worldwide. At the same time, conventional media distribution companies, including FOX, NBC, Discovery, etc., also notice this trend, and start to provide online video services to help people find and enjoy the world's premium video content whenever and wherever they want it. The success of Youtube and Hulu (Fig. 1-1) has shown the tremendous impact of digital video industry on our daily lives.

This rapid evolution of digital video industry has brought many new applications and techniques. Among them, video adaptation, video summarization and video compression are the rudimentary techniques. Sophisticated applications, like video indexing, browsing, archiving, cataloging, retrieving and rendering, are based on those fundamental ones. To date, the contradiction between the huge amount of video data and the limited manpower is an urgent problem to be solved. Thus, researchers and developers are seeking new approaches in these areas to improve the efficiency and reduce the cost of human manipulation.

1.1.1 Video Adaptation

Video adaptation is the process to fit and render a source video into a target display, when the resolution or/and aspect ratio of source and target are different. Due to the copious source resolutions and aspect ratios (Fig. 1-2), video adaptation is a fundamental and important technique, and is widely used in current consumer electronic

markets. Video adaptation can be performed without difficulty by applying some basic operations like center cropping, squeezing, or black-padding; the drawbacks of these approaches are obvious: fixed-window cropping will cause information loss on the frame boundary; squeezing will cause unpleasant geometric distortion; black-padding, without distortion and loss of boundary information, is a waste of the scarce display pixels. Meanwhile, as wide-screen video sources and displaying devices are gradually taking the place of the standard 4:3 videos and devices, some film companies employ humanbeings to manually adapt their films into many versions to fit different displays. This endeavor, is obvious tedious, inefficient and costly. Thus, there emerge a strong desire of an automatic video adaptation algorithm which can intelligently adapt to the video contents.

In recent years, content-aware video adaptation becomes a hot research topic that continually attracts researchers' interests. There exist two challenges for content-aware video adaptation. The first is how to establish a model to indicate the importance or saliency of video contents. The second is how to map the important contents such that users can simultaneously experience least information loss and best visual-friendliness.

1.1.2 Video Summarization

Video summarization is the process to extract an abstract of a video sequence and present it in a compact manner, that helps to enable a quick browsing of the video sequence and to achieve efficient content access and representation. Consider these situations: most people will be bored by an unedited and long home video captured by a non-professional; people may want to quickly "grab" the story progress of a TV series which has 40 episodes; or people would like to choose from a huge database of movies to quickly decide which one to watch. A tool that can automatically shorten the original video while intelligently preserve the most important and exciting video segments will be highly in demand.

Most of the earlier works do not consider the variation of video contents. By randomly or uniformly sample the original video, a simple summarization can be obtained. However, such an arrangement may fail to capture the real video content, especially when it is highly dynamic. It cannot remove the redundancy in the story, neither can it keep most salient segments. Thus, more sophisticated researches are conducted for seeking content-aware video summarization techniques. Fig. 1-3 shows an example that conventional summarization technique fails to capture dynamics of the video content. Like video adaptation, video summarization is also a challenging task for its highly subjective nature. Our research yields a promising solution for content modeling and video summarization.

1.1.3 Video Compression

Video compression, as the name suggests, is the technique to exploit the redundancy of a video sequence, thus to store or transmit it more efficiently. For decades, people have proposed many techniques for exploiting the redundancy both spatially and temporally by employing predictive coding, transform coding, entropy coding, etc. The state-of-art video compression standard, H.264/AVC, achieves huge rate-distortion gain over its preceding standards like H.263 and MPEG-2, by incorporating a number of sophisticated techniques including rate-distortion-optimization (RDO), quarter-pel motion estimation, etc. However, it still lacks the ability to adjust the coding parameters for various contents, thus sometimes becomes inflexible and sub-optimal. An example is shown in Fig. 1-4: Although people usually pay higher attention on the red region (human face) than the blue region (trees outside), they are treated equally in an H.264 encoder; and since the tree area has a rich texture, it may consume considerable coding bits. This is undesirable, especially when resources are limited. To overcome this problem, we need to address challenges such as content modeling and resource allocation. Our research provide a generic framework for

content-aware video coding. Also, an improved intra coding scheme is proposed to improve the coding efficiency of H.264 intra prediction.

1.2 Outline

The outline of this dissertation is presented as follows, along with a summary of our contributions in these areas: Chapter 2 presents content-aware video adaptation techniques. In this chapter, we first overview traditional video adaptation methods and representative available approaches for content-aware video adaptation techniques. Then, we address the two challenges of video adaptation respectively. For the first challenge – content modeling, we propose a nonlinear spatial-temporal saliency fusing approach that considers human perceptual characteristics. We incorporate features from both spatial and temporal domain. The saliency maps, which are indicators of content importance, are fused nonlinearly to imitate the human perception process. For the second challenge – content mapping, we propose to take both intra-frame visual considerations and inter-frame visual considerations into account, where intra-considerations focus on measuring the information loss within a frame, and inter-considerations emphasis the visual smoothness between frames. We segment the whole video in a shot/subshot basis, where the subshots are fixed-length groups of consecutive video frames. The boundary frames in a shot/subshot are intra-frames, for which a novel content-aware cropping and scaling metric is proposed and best mapping parameters are found by a hierarchical brute-force-search within that frame. For the rest of inter frames in the subshot, we minimize visual information loss accumulation under the constraint of visual consistency (inter-frame consideration). The optimization is formulated as a shortest-path problem in graph theory and we use dynamic programming to yield the temporal transition trace of cropping windows.

Chapter 3 presents content-aware video summarization approaches. In this chapter, we first overview some previous approaches of video summarization methods, both content-unaware and content-aware. Then, after analyzing the challenges of this

problem and limitations of existing works, we propose a novel approach to perform the video summarization task. This approach includes unsupervised learning of original video-audio concepts and hierarchical (both frame and shot levels) skimming. We first define an intermediate cognitive level term - concept primitive, to extract the structure of original video by concept primitive mining. Viewing it as a clustering problem, we propose to use bag-of-words (BoW) model for shot feature extraction, and use the scale-invariant-feature-transform (SIFT) to get the visual words and the matching-pursuit (MP) decomposition for generating the audio words, from both visual and audio sensory channels filtered with saliency masking, then cluster them into several groups by spectral clustering. Each cluster represents a certain concept primitive. Next, we summarize the original video from reconstruction point of view based on the learned concept primitives. While most researchers regard video summarization as a “subtraction” process, we regarded it a “summation” process. We propose a reconstruction reference tree (RRT) structure to efficiently represent the characteristics of a video sequence. Keeping at least one shot for each concept primitive, the concept integrity of summarized video is guaranteed to offer viewers the capability of context recovery. In addition, given a specified skimming ratio, we generate a video that also contains maximum achievable saliency accumulation (scalability). The summarization process is conducted in an iterative fashion, allowing flexible control of summarized video information richness vs. skimming ratio. Finally, to meet the skimming ratio specification and keep the smooth transition in the summarized video, we add a frame level saliency thresholding followed by a temporally morphological operation as post processing.

Chapter 4 presents content-aware video compression techniques. In this chapter, we briefly overview some existing works on this field, and propose a generic content-aware coding framework for low bit rate video applications. The key idea of this framework is to regard the content-aware coding as a resource allocation problem, where resources

are allocated between Intra and Inter frames within a group-of-picture (GOP) structure. We propose to treat different types of frames differently and thus adjust the quantization parameters correspondingly.

Chapter 5 presents a new intra coding algorithm to improve the H.264/AVC intra prediction. We propose to analyze the binary transition points on a block boundary to implicitly derive the prediction direction. Meanwhile, a new encoding order scheme is incorporated. The transition-based intra coding can achieve up to 10% bitrate savings over H.264 reference software JM. In addition, a novel video coding parameter analyzer is designed to assist the development of new coding algorithms for next-generation video compression standard. The design philosophy are also introduced in this chapter.

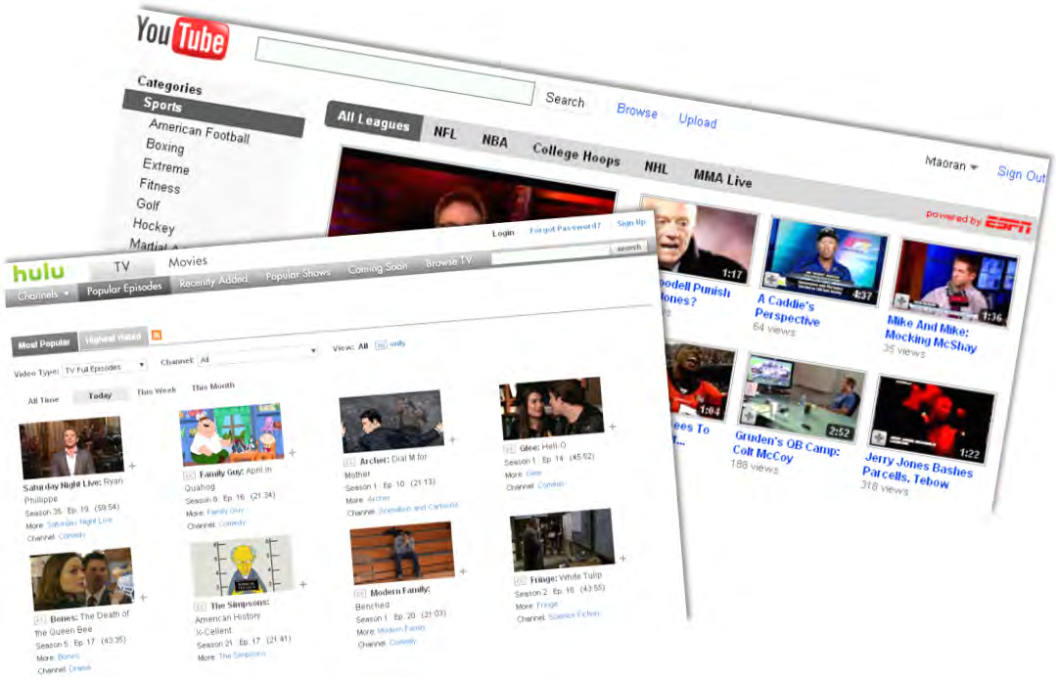


Figure 1-1. Successful online video service providers: Youtube and Hulu.

Figure 1-2. Copious digital video applications and conventional video adaptation methods.

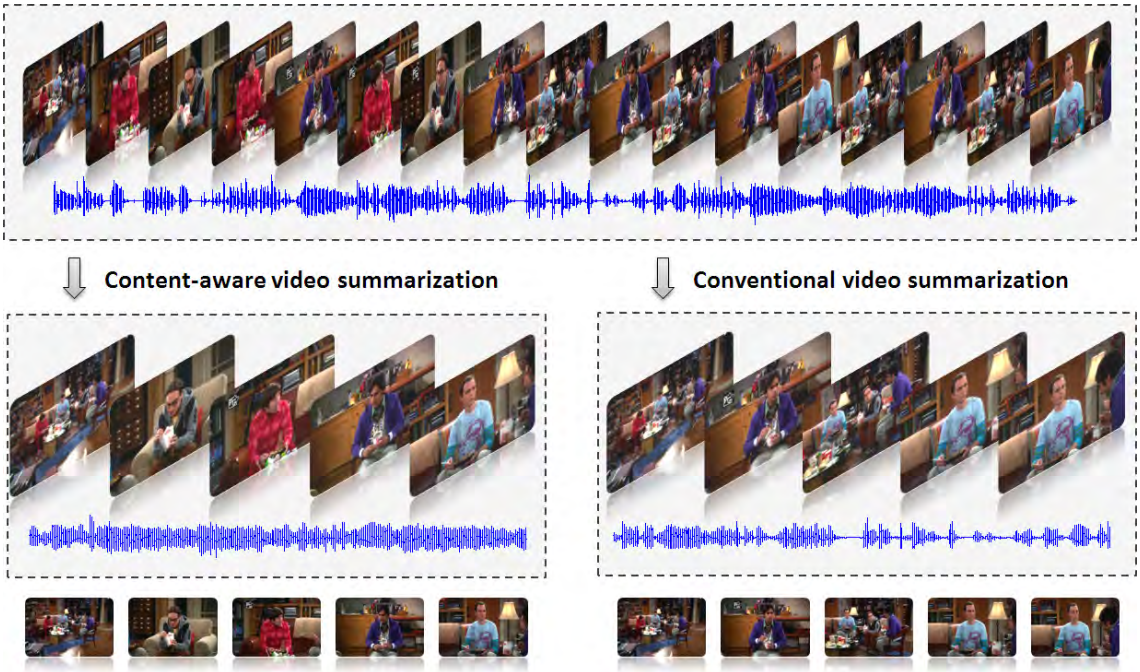


Figure 1-3. Content-aware video summarization vs. conventional video summarization. Bottom left: content-aware video summarization that extracts all concepts from original video and remove the structure redundancy. Bottom right: conventional video summarization by uniform sampling, which fails to cover all concepts from original video while still having redundancy.

Figure 1-4. Block based conventional video coding technique gives equal weight for every Macroblock. Red ellipse: Human face. Blue rectangle: Trees outside.

CHAPTER 2
CONTENT-AWARE VIDEO ADAPTATION

2.1 Introduction

Nowadays, the development of digital video applications has two opposite trends. On one hand, people savor the fantastic video contents delivered in cinemas and on high definition televisions (HDTVs) with resolution higher than 1920×1080. On the other hand, people enjoy the flexibility of playing videos on their portable devices (iPhone, BlackBerry, etc.) with resolution smaller than 480×320. These two trends evoke growing interests in automatic video adaptation that seeks to change the resolution (as well as aspect ratios), generally from the larger to the smaller, of video sources while faithfully conveying the original information. The task of adapting and re-rendering a video onto arbitrary display size or aspect ratio is termed as video adaptation, or video retargeting. A compelling retargeting algorithm aims at preserving the viewers' experience when the resolution and/or aspect ratio changes. Video retargeting is naturally a challenging problem, as it is a very subjective task to map human cognition into the automated process.

2.1.1 Content-Unaware Video Adaptation

A straightforward way to perform this task is to fit an original video into a target display via resizing with black-padding. When the aspect ratio of the target display varies from the original display, two black bands are padded on the boundary of target display to make the retargeted video free of geometric distortion. This approach, which is most often adopted in current consumer electronic market, cannot efficiently utilize the scarce screen pixels of a small device. Another naive approach is to crop the original display via a fixed window (generally located in the center of the frame) of the target display size. This approach will fully utilize the screen of the target device, but may cause the danger of losing important contents on the boundaries. Another approach is squeezing, when the aspect ratio of original and target devices are different, the frame

will experience some extent of geometric distortions, making uncomfortable feelings for viewers. Fig. 2-2 presents examples of the content-unaware retargeting methods.

2.1.2 Previous Approaches of Content-Aware Video Adaptation

Although content-unware video retargeting methods have the least computation costs and generally produce acceptable results, people are seeking more sophisticated solutions that can intelligently select the contents to present. However, there are two major challenges we have to deal with.

The first challenge for content-aware video retargeting approaches is: how to identify "important" contents? Although the human perception is a biological and phycological process that are not yet fully understood, studies on the user-attention model have shown that, such a model can be used to simplify the behaviors of the very complex human visual system. The studies, also referred to as saliency analysis, could be classified as the top-down (object or event driven), or the bottom-up, like feature-driven [1, 2] in literature. Features may come from both spatial and temporal domains. In the latter class, the spectral residue approach [2] suggests utilizing the spectral residue for an image to obtain the attention model. Later, Guo *et al.* showed that the phase spectral alone is good enough as a spatial feature, and they extend [2] for video by incorporating motion features under a phase spectrum of quaternion Fourier transform (PQFT) framework [3] (we call it baseline-PQFT). This work provides a new insight for a spatial-temporal saliency detection. However, putting the motion feature into a channel of a quaternion image lacks its physical interpretation, as the motion channel is actually a derivative image, which is not commeasurable with the other three color channels in original image domain. Besides, the naive motion feature obtained by frame differencing does not truly reflect human's perceptual experience: it's the local motion, not the global motion, that triggers the human interests mostly. Considering this, many people use optical flow based approaches [4] to factor out the global motion. While these methods are more reliable than the naive differencing, they have to deal with

the heavy computation burden and the aperture problem. Besides, those approaches usually adopt a linear combination scheme to fuse features in different domains, where the weighing factors need to be carefully selected. The saliency analysis will result in an output named “saliency map”, which represents the conspicuity (or saliency), at every location in the original image by a scalar quantity and to guide the attended locations. Given the saliency map, the content importance can be computed.

The second challenge for content-aware video retargeting approaches is: how to map important contents from the source to the target display? There are many possible solutions. To utilize entire target display efficiently, [5], [6], [7] proposed content-aware retargeting methods. Based on generated saliency map, they rearrange pixels in target frames: the original geometric layout are faithfully maintained for adjacent pixels with higher visual saliency, while other less salient pixels are morphologically squeezed to make up for the original-target display size difference. These methods work well on still images because viewers tend to concentrate on those salient areas and generally neglect other areas with little interests. They are most successful for images with natural sceneries. For example, curves (e.g. profile of a mountain, streams, trees) are robust to geometric distortion. However, for images with objects whose shapes can be precisely expected (like buildings), these methods become disastrous as their anamorphic strategy often results in unjustified object shape distortion. In order to avoid this problem, single frame optimization [4] [8] methods are proposed to apply a cropping window to pan throughout each original frame to yield a region of interest with the degradation inside minimized. Visual consistency is claimed by smoothing optimal cropping window parameters of each frame. This endeavor, nevertheless, does little help to remove visual inconsistency along temporal axis because the cropping window parameters of adjacent frames are optimized independently, and many twists and turns still exist on the window trace after smoothing. This suggests obvious frame jump back and forth, zoom in and out in the targeted video, which leads to viewer

vertigo very soon. To carefully consider the visual experience along temporal axis, a back-tracing method [9] is presented to dynamically determine the cropping window trace. This method adds another constraint to bound possible shift of cropping windows among adjacent frames when optimizing the window trace. It produces a retargeted video with frame inconsistency removed and thus avoids viewers' discomfort. However, This method unfairly favors the initial location of cropping window and clamp sequent cropping window locations near the initial value. Thus, the window cannot crop salient objects soon as frame goes further, when the location of salient objects is quite different from that of the first frame. Consequently, this method cannot handle videos with frequent content change. Some examples of existing content-aware retargeting methods are shown in Fig. 2-3.

2.1.3 Overview of the Proposed Approach

We address the two challenges respectively. For saliency detection, we propose a novel spatial-temporal saliency map based on nonlinear fusion. In our scheme, spatial saliency is detected by the phase spectrum of quaternion Fourier transform (PQFT) on a color image (video frame), which utilizes the multiple image channels as a vector field to exploit conspicuous spatial features (color, intensity, etc.); motion saliency is measured by local motion (global motion residue), where the global motion parameters are estimated by robust affine fitting with least median of squares (LMedS) [10], from a set of matched feature points detected by the Kanade-Lucas-Tomasi (KLT) feature tracker [11]. Unlike the dense optical flow approaches, the KLT tracker works on sparse feature points, and thus is more efficient. Then, the spatial and temporal saliency maps are nonlinearly fused. The innovation for this nonlinear fusion is based on human perceptual properties: 1) When excitation is absent (texture uniformly distributed), people tend to focus on the center of the frame, instead of the borders. 2) The human perception process consists of a "stimulating" phase and a "tracking" phase, defined as "saccade and pursuit" [12] in human vision theory. First, spatial-salient regions pop up

as primitive “stimulus”. If a spatial-salient region has significant local motion activities, this motion stimulus will strengthen the spatial stimulus, and cause higher attention. Otherwise, “lazy” human eyes will continually focus on the spatial-salient regions. 3) Occasionally, spatial and motion saliency regions are not consistent. Our scheme treats this as a “prohibited case” since the motion stimulus will distract the spatial stimulus and make a rapid change of focus points, which will cause an eye fatigue. Thus, professional photographers and moviemakers will make their efforts to avoid the situation.

For content mapping, we propose a framework to adapt real life videos, which can be as long as an entire movie rather than merely a clip. Our framework is originated from the cropping-and-scaling method developed by [4], but with new features. Note that viewers are not sensitive to abrupt cropping window change for adjacent frames with rapid scene change at shot boundaries. We first detect shots [13] [14] and then perform the task independently. A shot is then decomposed into subshots for visual comfort and computational efficiency. For each frame, a 3-parameter (scale and location) rigid cropping window is determined to select a region of interest as retargeted frame. Within a shot, we propose a motion-prediction method to find an optimal fixed scale for the cropping window as otherwise a mild scale variation may cause significant visual degradation. Regarding the optimal location of cropping window, we first process the two boundary frames of each subshot. Aiming at keeping as much fidelity to original frame as possible, a cropping window is selected to minimize an information loss function due to cropping and resizing and yield source and destination locations of cropping windows in the subshot. For other frames within, we address viewer visual expectations as contradictory intra-frame “fidelity” and inter-frame “visual-inertness” and minimize an accumulative loss function including both information loss accumulation and visual-inertness loss accumulation. Then the dynamic trace of rigid cropping windows from source to destination locations is optimized as a shortest path problem with dynamic programming solution. As subshot alternates, the destination locations are

updated to synchronize content of the frames at that time. Thus as frame goes further, the cropping window is not clamped onto the adjacency of source location as in [9] and is still capable of cropping salient objects of interest. Our approach can be applied to any type, any length of video, no matter how fast the video content changes. Our retargeting results are free of shape distortion, have no annoying zoom in/out artifacts within the same scene, preserve the salient objects of interest throughout and keep visual consistency as well. The computational load of our method includes the brutal force search at boundary frames and dynamic programming for other frames. Besides, with C++ implementations, our approach can perform the task in real time.

Our innovation points also include content-aware information loss metrics, and a hierarchical search to find optimal retargeting parameters on a single frame. Compared to the content-independent scaling penalties [4], our metric can adjust the scaling factor corresponding to different contents. our scaling metric also outperforms the content-aware scaling metric in [8], as we take into account not only the anti-aliasing filter in [8] but also the true resizing process. The hierarchical search, can greatly save computation costs.

Fig. 3-1 and Fig. 2-5 illustrate the procedure of our approach. Fig. 3-1 shows the saliency detection and optimal signal frame adaptation by finding best cropping window parameters, and Fig. 2-5 shows the dynamic programming approach for optimizing the trace of the cropping window dynamically. In the following sections, we will discuss our content modeling and content mapping algorithms in detail.

2.2 Spatial-Temporal Saliency Map

2.2.1 Spatial Saliency Map

Denote the *n-th* frame in the video sequence F^n. The frame can be represented as a quaternion image [15] which has four channels, $q^n = Ch_1^n + Ch_2^n\mu_1 + Ch_3^n\mu_2 + Ch_4^n\mu_3$, where μ_i, $i = 1, 2, 3$ satisfies $\mu_i^2 = -1$, $\mu_1 \perp \mu_2$, $\mu_2 \perp \mu_3$, $\mu_1 \perp \mu_3$, $\mu_1 \perp \mu_2$. Ch_j^n, $j = 1, 2, 3, 4$ are the channels of the quaternion image. If choosing μ_1 along the luminance axis, i.e.,

$\mu_1 = (i+j+k)/\sqrt{3}$, the color image is thus decomposed into luminance and chrominance components Y^n, C_b^n and C_r^n, and the quaternion image is pure ($Ch_1 = 0$) [15]. We can further represent q^n in *symplectic* form: $q^n = q_1^n + q_2^n\mu_2, q_1^n = Ch_1^n + Ch_2^n\mu_1, q_2^n = Ch_3^n + Ch_4^n\mu_1$. The quaternion Fourier transform (QFT) of the quaternion image $q^n(x,y)$ can be calculated by two complex fourier transforms of the *symplectic* parts: $Q^n[u,v] = Q_1^n[u,v] + Q_2^n[u,v]\mu_2$. The forward and inverse fourier transform of each part are:

$$Q_i^n[u,v] = \frac{1}{\sqrt{WH}} \sum_{y=0}^{W-1} \sum_{x=0}^{H-1} e^{-\mu_1 2\pi(\frac{yv}{W}+\frac{xu}{H})} q_i^n(x,y) \tag{2–1}$$

$$q_i^n(x,y) = \frac{1}{\sqrt{WH}} \sum_{v=0}^{W-1} \sum_{u=0}^{H-1} e^{\mu_1 2\pi(\frac{yv}{W}+\frac{xu}{H})} Q_i^n[u,v] \tag{2–2}$$

where (x,y) is the spatial location of each pixel,W and H are image's width and height, and $[u,v]$ is the frequency.

The phase spectrum of $Q^n[u,v]$ (Q for abbreviation) can be calculated by $Q_P = Q/||Q||$. Taking the inverse transform of the phase spectrum Q_P as in Eq.(2–2), the spatial saliency map is obtained by smoothing out the squared L_2 norm of q_P with a two-dimensional Gaussian smoothing filter g.

$$SM_s = g * ||q_P||^2 \tag{2–3}$$

The advantages of the PQFT approach over traditional multi-channel phase spectrum of the 2D Fourier transform (PFFT) is shown in Fig. 2-6. PQFT not only achieves better saliency detection results by treating a color image as a vector field, but also consumes less computation time since only two complex 2D Fourier Transforms are conducted for the *symplectic* equations, while PFFT has three (one for each channel).

2.2.2 Motion Saliency Map

There are two steps to obtain the motion saliency map: a) Kanade-Lucas-Tomasi (KLT) tracker to get a set of matched good feature points [11] and b) robust affine parameter estimation by least median squares (LMedS) [16]. Denote the displacement of a point $\mathbf{x} = (x,y)^T$ at previous frame F^{n-1} to current frame F^n as $\mathbf{d} = (d_x, d_y)^T$, a

six-parameter affine model is adopted to estimate the global motion: $\mathbf{d} = \mathbf{D}\mathbf{x} + \mathbf{t}$, where $\mathbf{t}$ is the translation vector $\mathbf{t} = (t_x, t_y)^T$ and $\mathbf{D}$ is a 2×2 deformation matrix. Good features are located by checking the minimum eigenvalue of each gradient matrix, and the good features are tracked using a Newton-Raphson method. The point $\mathbf{x}$ in F^{n-1} moves to point $\mathbf{x}$'=$\mathbf{A}\mathbf{x} + \mathbf{t}$ in F^n, where $\mathbf{A} = \mathbf{I} + \mathbf{D}$ and $\mathbf{I}$ is a 2×2 identity matrix. The model parameters are estimated by minimize the dissimilarity in each feature window W:

$$\varepsilon = \iint_W (F^n(\mathbf{A}\mathbf{x} + \mathbf{t}) - F^{n-1}(\mathbf{x}))^2 \omega(\mathbf{x}) d\mathbf{x} \tag{2–4}$$

We adopt the least median squares to estimate the affine parameters robustly [16]. The global compensated image is generated by warping with the estimated $\hat{\mathbf{A}}$ and $\hat{\mathbf{t}}$. The absolute difference of the original frame with its global-compensated version is used to generate the motion saliency map.

$$SM_m = g(\mathbf{x}) * |F^{n-1}(\mathbf{x}) - F^n(\hat{\mathbf{A}}^{-1}[\mathbf{x} - \hat{\mathbf{t}}\ ])| \tag{2–5}$$

2.2.3 Nonlinear Fusion of Spatial-Temporal Saliency Map

When both the spatial and temporal saliency maps are available, the final saliency map is generated by a spatial-masked nonlinear manner which imitates the human vision features. First, a 2D Gaussian layer G centered at the frame center is fused to the spatial saliency map: $SM_{spatial} = SM_{spatial} \times G$. A binary mask M_s of spatial saliency significance is generated by thresholding. The final saliency map SM is obtained by:

$$SM = \max(SM_s, SM_m \cap M_s) \tag{2–6}$$

The reasons to use the $\max$ operator with a binary masked motion saliency map and a Gaussian layer are many. 1) Gaussian layer is used to adjust the descending importance from the center of a frame to the border. 2) Mask is used to exclude the spatial-temporal inconsistent cases (the “prohibited cases”). 3) The mask enhances the robustness of the spatial-temporal saliency map, when global-motion parameters are not

estimated correctly. 4) The $\max$ operation avoids the depression of insignificant salient regions caused by renormalization if using a linear combination scheme. 5) The $\max$ operation avoids the selection of weighting factor between spatial and motion saliency maps. Fig. 2-7 shows the comparison of linear combination, naive nonlinear (unmasked $\max$) fusion and our scheme, where in the first video the global motion parameters are correctly estimated but in second video are wrong. The comparison shows the robustness of our scheme.

2.3 Intra-Frame Visual Consideration

Our retargeting framework starts at single frame adaptation on the first and last frames in a subshot. In this phase, we only consider intra-frame information loss, and our target is to minimize the information loss caused by retargeting. Thus, the first important step is to objectively model the information losses within one frame. We call this the intra-frame visual consideration.

2.3.1 Content-Aware Information Loss Metrics

To better quantify the information losses within a frame, we consider both the saliency loss due to cropping and the scaling penalty. We propose a content-aware information loss metric L, which consists of two terms. The first term L_c is content-aware cropping loss and the second term L_s is content-aware scaling loss.

$$L = (1-\lambda)L_c + \lambda L_s \tag{2–7}$$

$$L_c = 1 - \sum_{(x,y)\in W} SM(x,y) \tag{2–8}$$

$$L_s = \sum_{(x,y)\in W} (F(x,y) - \hat{F}(x,y))^2 \tag{2–9}$$

where SM is normalized such that $\sum_{(x,y)} SM(x,y) = 1$, W is the cropping area, and $\hat{F} = upsizing(g * downsizing(F,s),s)$. λ is a factor to balance the importance of cropping and scaling, it is adjustable to user's preference. The proposed L_s better measures the scaling loss for different contents than [4, 8], by taking into account the real resizing

process. Integral images are used for computing the L_c and L_s by a look up operation on integral images.

2.3.2 Hierarchical Search for Optimal Single-Frame Cropping Window

Under a cropping-scaling framework, the retargeting optimization for a single frame is to find best parameters (x, y, s) with minimum information loss, where (x, y) is the location of the top-left point of the cropping window over original frame, and s is the scaling factor. Isotropic scaling is used to avoid geometric distortion. After the information loss metric is well formulated, a hierarchical brute-force search is used to find the best retargeting window parameters $(\hat{x}, \hat{y}, \hat{s})$.

$$P(\hat{x}, \hat{y}, \hat{s}) = \arg\min_{x,y,s} L(x, y, sW_t, sH_t) \tag{2–10}$$

Note the search range of (x, y) is constrained by s, and s is constraint by $1 \leq s \leq min(W_s/W_t, H_s/H_t)$, where W_s, W_t, H_s, H_t are the width and height for source and target frames, respectively. The searching range is a group of surfaces in x, y, L space. Fig. 2-8 shows the searching space. Each surface corresponds to a particular scaling factor s. The point which yields the minimum L (the lowest point in space) is the best parameter (x, y) and the surface it belongs to is the best s.

For computation saving purposes, the search for (x, y, s) is first on a coarse (x, y) grid (a 10×10 grid search will save 99% computation over 1×1 grid search), a target parameter set (x_1, y_1, s) is found after the coarse search. The second search is a fine search within a range around (x_1, y_1) with the fixed scaling factor s found previously. After the hieratical search, the best parameter set (x_2, y_2, s) is obtained.

Considering that humans are very sensitive to scale variation even with a modest value, we alternatively determine a good scale s using the method described in Sec. 2.4.2. The scale parameter is then fixed s throughout a shot. The three-dimensional

optimization problem is reduced to a two-dimensional search for optimal x, y instead.

$$P(\hat{x}, \hat{y}) = \arg\min_{x,y} L(x, y, \hat{s}W_t, \hat{s}H_t) \tag{2–11}$$

2.4 Inter-Frame Visual Consideration

A significant difference of video retargeting task from resizing a still image is temporal considerations. i.e., if we only minimize intra-frame visual information loss, the resultant video still suffers from annoying jitters due to independent but inconsistent window parameters. Here we also take into account the fact that viewers need a steady and smooth video content transition known as "visual inertness" requirement. Note that across adjacent frames, a shift of cropping windows imposes artificial camera motion to retargeted frames. On one hand, an absolute free inter-frame shift makes it possible to crop and then preserve the most salient region of each different frame. On the other hand, visual inertness favors a modest inter-frame shift or no shift at best. In our approach, we consider these contradictory visual comfort clues together and optimize their total value. To measure the visual performance with respect to the location of cropping window, we define a function of visual penalty accumulation within a subshot as in Eq. 2–12.

$$\begin{aligned} Q(\bar{x}_N, \bar{y}_N) &= \sum_{i=1}^{N} L(x_i, y_i) + \omega \cdot \sum_{i=2}^{N} EI(x_{i-1}, y_{i-1}, x_i, y_i) \\ EI(x_{i-1}, y_{i-1}, x_i, y_i) &= (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \end{aligned} \tag{2–12}$$

where L is intra-frame visual information loss of frame i, EI is temporal penalty that constrains shift of panes across adjacent frames. x_i, y_i is the location of upper-left corner of cropping pane of frame i and N is the total number of frames in a subshot. $(\bar{x}_N, \bar{y}_N)$ is a dynamic trace of the upper-left corner of the cropping window over a subshot. Our goal is to find the optimal trace $(\hat{\bar{x}_N}, \hat{\bar{y}_N})$ such that Q is minimized.

2.4.1 Dynamic Programming Solution for Optimization of Cropping Window Parameters

We model the solution space $(\bar{x}_N, \bar{y}_N) = \{x_i, y_i\}_{i=1}^N$ by a graph illustrated in Fig. 2-9, where each node (x_i, y_i) denotes the upper-left corner location of a candidate cropping window of frame i and each edge $(x_{i-1}, y_{i-1}) \rightarrow (x_i, y_i)$ represents the shift of cropping window from frame i-1 to frame i. The cost on each node is visual information loss $L(x_i, y_i)$ and as for each edge, the cost corresponds to temporal penalty $EI(x_{i-1}, y_{i-1}, x_i, y_i)$. Thus minimizing Q in Eq. 2–12 is equivalent to finding the shortest path from node (x_1, y_1) to (x_N, y_N). The optimization can be easily solved by dynamic programming (DP). The recursive format of the objective function is given in Eq. 2–12:

$$\begin{aligned} Q(\bar{x}_i^k, \bar{y}_i^k) = \min_j \{ & Q(\bar{x}_{i-1}^j, \bar{y}_{i-1}^j) + \\ & \omega \cdot EI(x_{i-1}^j, y_{i-1}^j, x_i^k, y_i^k)\} + L(x_i^k, y_i^k) \end{aligned} \tag{2–13}$$

where $Q(\bar{x}_1^1, \bar{y}_1^1) = 0$, $Q(\bar{x}_i^k, \bar{y}_i^k)$ denotes minimized cost accumulation or equivalently the shortest path from source node (x_1^1, y_1^1) of frame 1 to the kth node of frame i. $Q(\bar{x}_{i-1}^j, \bar{y}_{i-1}^j)$ is the shortest path up to the jth node of frame i-1, $EI(x_{i-1}^j, y_{i-1}^j, x_i^k, y_i^k)$ denotes the cost of edge connecting the jth node with frame i-1 to the kth node of frame i and $L(x_i^k, y_i^k)$ is the cost of the kth node of frame i. Algorithm 1 presented the algorithm to find the shortest path between the source and destination nodes.

The remaining question is how to choose two boundary nodes as the source and destination between which a shortest path is searched. As mentioned before, a shot is divided into equal-length subshots. We assign destination as the location of optimized cropping window in Eq. 2–11 and source as the location of the destination of last previous subshot to avoid jitter between subshots. By this measure, every subshot, we update cropping window to a free position that crops most salient area of original frame and preserve most visual information at that time. Meanwhile, dynamic programming method yields a cropping window transition trace from the start to end of the subshot, with consideration of both least visual information loss and least visual inconsistency. So

input : Source node of cropping window (x_1, y_1), Destination node of cropping window (x_N, y_N) and video frames $\{I_i\}_{i=1}^{N}$ of the subshot, the number of nodes $C[i]$ of frame i

output: Optimal trace $(x_{\hat{N}}, y_{\hat{N}})$ as the shortest path from source to destination

```
x_1^1 ← x_1, y_1^1 ← y_1, Q(x̄_1^1, ȳ_1^1) ← 0, (x̄_1^1, ȳ_1^1) ← (x_1^1, y_1^1);
for i ← 2 to N do
    extract from video the ith frame in subshot as I[i];
    calculate saliency map SM[i] of I[i];
    for k ← 1 to C(i) do
        calculate cost of Node (x_i^k, y_i^k) as L(x_i^k, y_i^k) ;
        T_opt ← ∞ ;
        for j ← 1 to C(i − 1) do
            calculate Edge cost EI(x_{i−1}^j, y_{i−1}^j, x_i^k, y_i^k);
            T(x̄_i^k, ȳ_i^k, j) ← Q(x̄_{i−1}^j, ȳ_{i−1}^j) + ω · EI(x_{i−1}^j, y_{i−1}^j, x_i^k, y_i^k) ;
            if T(x̄_i^k, ȳ_i^k, j) < T_opt then
                BackPt[(x̄_i^k, ȳ_i^k)] ← (x̄_{i−1}^j, ȳ_{i−1}^j) ;
                T_opt ← T(x̄_i^k, ȳ_i^k) ;
            end
        end
        Q(x̄_i^k, ȳ_i^k) ← T_opt + L(x_i^k, y_i^k) ;
    end
end
for k ← 1 to C(N) do
    T(k) ← Q(x̄_N^k, ȳ_N^k) + ω · EI(x_N^k, y_N^k, x_N, y_N);
end
k̂ ← arg min_k T(k), (x̂_N, ŷ_N) ← (x_N^k̂, y_N^k̂) ;
// tracing back
for i ← N − 1 to 2 do
    (x̂_{i−1}, ŷ_{i−1}) ← BackPt[(x̂_i, ŷ_i)]
end
(x_N̂, y_N̂) ← {(x̂_1, ŷ_1) ← ··· ← (x̂_N, ŷ_N)}
```

Algorithm 1: Proposed algorithm for dynamic cropping window trace searching.

our method can always crop salient region no matter how frequent change occurs in a shot.

2.4.2 Optimize Scale in A Shot

The optimal scale which minimizes information loss function depends on weight λ. Generally, λ is specified according to aesthetic preference of viewers and it is quite subjective among different viewers. Most scenes of movies, news or commercials

portrait foreground salient objects in high resolution. So we assume that most viewers are more expecting a cropping window with complete objects in as people usually prefer a global view with broad visible range at the price of resolution rather than only access to a limited area. Here resizing is more preferred to cropping. On the contrary, in most long distance shot scenes (e.g. sport broadcasting) where salient objects occupies small areas, most viewers would like to focus on and track the object without huge resolution degradation, otherwise, objects become too small to recognize. Here cropping becomes more preferable to resizing.

Based on aesthetic requirement, we specify an initial weight λ and find optimized scales of some sampled frames based on Eq. 2–10 and average them as the scale of the shot. Mostly, this simple method works fine, however, when a salient object is moving fast, the cropping window may not move fast enough to catch up with the object due to visual consistency constraint. This leads to cut off some parts of the object and it suggests a larger scale of cropping window needed. We use the velocity of dynamic cropping window transition within a shot to estimate how fast objects of interest move. Then based on the velocity estimate, we adjust the weight λ in order to obtain a larger scale, which yields a larger cropping window to include salient objects completely.

$$\lambda' = (1 + \exp\{-|\frac{1}{N} \cdot \sum_{i=1}^{N} \frac{(\hat{x}_i - \hat{x}_{i-1})^2 + (\hat{y}_i - \hat{y}_{i-1})^2}{L_i^2} - v_\alpha|\})^{-1} \tag{2–14}$$

where $\frac{1}{N} \cdot \sum_{i=1}^{N} \frac{(\hat{x}_i - \hat{x}_{i-1})^2 + (\hat{y}_i - \hat{y}_{i-1})^2}{L_i^2}$ is the velocity estimate of cropping window transition, N is the total number of frames in the shot. L_i is the maximum distance a cropping window can move from (x_{i-1}, y_{i-1}) and v_α denotes a reference velocity. Given the updated weight λ', a new scale average is optimized for the shot. Then we start over to find optimal trace of cropping window under the new scale.

2.5 Experimental Results

We implement our scheme in C++ with OpenCV (http://opencv.willowgarage.com/wiki/), FFTW3 (http://www.fftw.org/) and KLT (http://www.ces.clemson.

edu/~stb/klt/) libraries, and design a series of experiments to test the performance of our approach compared to the representative existing work on saliency modeling and retargeting. The experiments are carried out on various types of videos, including movie, entertainments, news, sports, etc. Original videos can be in any size and length. Viewers are allowed to specify any retargeted display size or customized aspect ratio. We evaluated our content-aware video adaptation framework by evaluating the performance of our content-modeling algorithm first. Then, we evaluate the proposed content-mapping algorithm with the saliency maps given. We present the experimental results in two ways. Since content modeling and video adaptation are highly subjective tasks, results are generally shown on figure illustrations without objective performance quantification. Therefore, our first way to present the experimental results is to use figure illustrations, like most of the existing work did to show their results. In addition, we go beyond the previous works by conducting subjective tests on the experimental results to collect mean opinion scores (MOS), which play the role to objectively evaluate the algorithms' performances. By conducting statistical analysis on the MOS of the results, we are able to draw a conclusion by hypothesis testing to quantify the performances of algorithms with a certain confidence level. We then record the hypothesis testing decision as the second way to present the experimental results.

2.5.1 Figure Illustrations

In the following, we present the experimental results with figure illustrations. The first illustration, Fig. 2-13, shows the content modeling results (saliency maps) generated by human, by saliency toolbox (STB, http://www.saliencytoolbox.net/) and by our content modeling algorithm respectively. We use a collection of images with multiple resolutions and aspect ratios for the experiments. Besides the saliency maps, we also illustrate the so called "proto-regions", which are found by a thresholding method in [2] using the saliency maps, to show the contours of salient regions on original images, as shown in the red-circled regions on original images in Fig. 2-13. By comparing the

saliency maps generated by STB and our algorithm with manually generated labels (generated by us as viewers), we may tentatively claim that the proposed algorithm outperforms STB as it better approximates humans' viewing experiences. For example, for the fourth image which shows two children playing at beach with a sailing boat in the sea, our algorithm successfully extracts the regions of children and the boat, which are also the focused regions of human eyes. The STB, on the contrary, is only able to capture the line between the sea and the sky. Later, we conduct subjective tests and statistical analysis to validate our claim. The details are presented in Sec. 2.5.2.

The second illustration, Fig. 2-10, shows the performances of image adaptation algorithms. Regarding image adaptation as a single-frame video adaptation, we compare the performance of our adaptation algorithm with two representative image retargeting algorithms: bidirectional similarity (BS) [7], which is a patch-based approach that maximize the similarity score of patches from source and target images bidirectionally, and seam carving (SC) [5], which aims at maximizing the total energy on the target image by removing low-energy seams on the original image. We use the test images from [7]. We can clearly see that our adaptation algorithm most faithfully keeps the geometric shapes of objects on original while presenting the most salient regions - the dolphin, the building and the house. SC and BS have to bear with geometric distortions, which are not desirable.

The third illustration is Fig. 2-11, which presents the comparison of video adaptation by our content mapping algorithm, with our saliency map and the saliency map generated by baseline-PQFT [3], respectively. Although our saliency map is an extension of the baseline-PQFT and shares a lot of similarities with it, the saliency maps of our method show a better performance to incorporate the motion features, as are illustrated in Fig. 2-11 (the moving human faces, tennis player and bee). Therefore, our content modeling algorithm can benefit from the more accurate saliency maps, and then result in better adapted videos.

The last illustration in this section is Fig. 2-12. It serves the purpose of showing the visual-friendliness characteristic of our adaptation algorithm, compared to two state-of-the-art content-aware video adaptation methods: single frame smoothing (SFS) [4] [8] and back tracing (BT) [9]. Generally, SFS suffers from jittering, which will cause uncomfortable feelings of viewers. Back tracing is mostly acceptable, however, the adapted video is not always able to preserve salient regions of interest in the original video. In comparison, our method throughout preserves salient region as frame goes further and avoids jitter effects as well. Fig. 2-12 presents result comparison in a static fashion. We illustrate cropping windows on original frames with frame number noted in Fig. 2-12. Original video is in resolution 640×352 and the specified retargeted size is 320×240. An initial weight λ (cropping/resizing preference) is provided as 0.3 and the subshot length is 120 frames. In results of SFS, although lion and zebra is preserved completely, the cropping window shifts back and force frequently, which means huge jitter effects in retargeted video. In results of BT, from frame $\#238$ to $\#259$, the cropping window includes complete zebra, however, as frame goes to $\#294$ and $\#318$, it is left behind by the zebra due to fast motion. So most parts of zebra is lost in retargeted video. In contrast, our result yields a visual consistent cropping window trace to preserve zebra completely. In order to make a convincing conclusion that our algorithm outperforms BT, we conduced subjective tests on adapted video sequences by our algorithm and BT. The details are presented in Sec. 2.5.2.

2.5.2 Subjective Evaluations

As we mentioned at the beginning of this section, we go beyond the previous works on presenting experimental results by conducting subjective tests and statistical analysis on the results. We carry out subjective evaluations in the form of online survey. In this section, we present the subjective tests in detail.

The first subjective evaluation is on the saliency maps in Fig. 2-13. The purpose of this test is to quantitatively measure that if our algorithm outperforms the STB on

saliency map generating, within a certain confidence interval. We set up a website (http://www.mcn.ece.ufl.edu/public/subjective/) to provide the testing materials for the experiment. On the website, we describe the purpose of this test and explain how to evaluate the saliency maps. For each of the nine original images, two saliency maps generated by our content-aware algorithm and the STB are both presented to the participants. Through out this evaluation process, the participants are blind to the name of the algorithms so as to avoid possible bias. Then, each participant are instructed to give each saliency map a score ranging from 1 to 5, 1 for the worst and 5 for the best, to indicate how good it is. The participants need not to make hasty decisions as they could take their time to make careful comparisons until they are confident about the score they have in mind.

After that, we collect the mean opinion scores (MOS) submitted by 60 participants and conduct statistical analysis on the MOS. We propose to use hypothesis testing, which is a popular statistical method to scientifically evaluate if there is enough statistical evidence to make a decision on a hypothesis with the experimental data. Since each participant evaluates the saliency maps of both algorithms, the paired-samples student t-test is considered most suitable for the analysis. The assumption of the experimental data is that the scores of the two saliency maps by each algorithm for image i follow normal distributions with the same variance and difference mean μ_1^i and μ_2^i, where μ_1^i is the mean of the scores for our algorithm. We make the null hypothesis as $H_0 : \mu_1^i \leq \mu_2^i$, which is interpreted as that our algorithm is not better than the STB; the alternative hypothesis is $H_1 : \mu_1^i > \mu_2^i$, which is interpreted as that our algorithm outperforms the STB. We take the significance level $\alpha = 0.05$, so the confidence level of our tests is 95%. We calculate the p-value for each test, and then make decision to reject the null hypothesis if the p-value is smaller than or equal to the significance level α.

The results of hypothesis testing are summarized in Table 2-1. Decision 1 means to reject the null hypothesis. It is shown that for all the nine images, we

Table 2-1. Hypothesis testing results for subjective evaluation on retargeting algorithms
1: Our approach 2: Saliency Toolbox (STB)
$\alpha = 0.05$, $df = 59$, t(critical one-tail)$= 1.671$

		Mean	Variance	t Stat	p($T_i = t$)	Decision
Img1	1	3.450	0.930	2.755	0.004	1
	2	2.967	0.880			
Img2	1	3.517	0.754	7.521	0.000	1
	2	2.200	0.908			
Img3	1	3.650	0.731	8.050	0.000	1
	2	2.217	0.918			
Img4	1	3.533	0.821	7.800	0.000	1
	2	2.083	0.959			
Img5	1	3.300	1.061	1.691	0.048	1
	2	2.950	1.031			
Img6	1	3.883	0.884	11.697	0.000	1
	2	1.983	0.830			
Img7	1	3.583	1.332	5.319	0.000	1
	2	2.483	1.034			
Img8	1	4.050	0.582	10.619	0.000	1
	2	2.067	0.945			
Img9	1	3.567	1.055	3.587	0.000	1
	2	2.883	1.223			

reject the null hypothesis that our algorithm is not better than the STB. Thus, we can make a conclusion that our algorithm consistently performs better than the STB with $95\% confidence$. Fig. 2-14 is the bar chart that depicts the scores' statistics for saliency maps by the two algorithms. Except image 5, the other eight images exhibit a consistency that the confidence intervals of STB and ours do not overlap. The results shown on Fig. 2-14 comply well with the hypothesis testing results from Table 2-1. We also have some interesting observations from Fig. 2-14. For example, the worst saliency maps generated by the STB are for image 4,6 and 8, while these are in real the most dissimilar ones with the manually generated labels. And for image 1, 5 and 9, where the STB and ours have the closest performances, the two saliency maps and our manually label resemble. This phenomenon shows that the participants have similar viewing experiences with us.

The second subjective test we have carried out is to evaluate the performance of our retargeting algorithm by comparing the output video sequences generated by our algorithm with outputs generated by other schemes. In practice, it is hard to generate or collect video demos of other video retargeting algorithms since most literatures presented their results on the form of papers. In order to make a fair and convincing comparison, we made efforts to contact the authors, asking for their help on the experiments. Fortunately, we got one reply from Dr.Thomas Deselaers in RWTH Aachen University, GE, that he would like to release his source code for their algorithm back tracing (BT) proposed in CVPR 2008 [9]. With his kindly help and guidance, we are able to compile and run their retargeting algorithm and generate outputs. The output videos are generated under optimal parameters with his guidance. Considered the interestingness of the subjective test and to reduce the burden of the participants, we pick four most interesting video clips to the test group: Avatar, Up, Madagascar and SpainTorres, and provide all the testing materials on our subjective testing website. For each clip, the participant are instructed to first watch the original version. Then, the participant could feel free to play the outputs by algorithm A and algorithm B (they were blind to the names to avoid bias) in any order or with arbitrary times, and give a score ranged from 1-5 when they are confident about the score, as they did for the previous test. We collect the final scores from 60 subjects, and analyze the statistical significance with hypothesis testing. In this case, the null hypothesis is that our algorithm will not outperform the BT.

The results of hypothesis testing are summarized in Table 2-2 and the statistics of scores are depicted in Fig. 2-15. Unlike the previous test, the hypothesis testing does not show the consistency to reject all null hypotheses for the four test sequences. We can also observe that 95% confidence intervals of the MOS scores of BT and our algorithm overlap. Thus, we cannot draw a safe conclusion that our adaptation

algorithm is statistically better than BT. We would say the two algorithms have similar performances while ours tends to be slighter better for a higher mean.

Table 2-2. Hypothesis testing results for subjective evaluation on retargeting algorithms
1: Our approach 2: Back tracing (BT)
$\alpha = 0.05$, $df = 59$, t(critical one-tail)$= 1.671$

		Mean	Variance	t Stat	p($T_i = t$)	Decision
Avatar	1	3.833	0.480	1.787	0.040	1
	2	3.583	0.722			
Up	1	4.067	0.640	4.373	0.000	1
	2	3.575	0.605			
Madagascar	1	3.750	0.767	0.186	0.427	0
	2	3.725	0.605			
SpainTorres	1	3.767	0.860	1.684	0.049	1
	2	3.542	0.757			

2.6 Summary

In this chapter, we have proposed a nonlinear approach to fuse the spatial and temporal saliency maps for video retargeting considering the human vision characteristics. We also presented the new content-aware information loss metrics and a hierarchical search scheme under a cropping and scaling retargeting framework. Meanwhile, a dynamic programming solution is proposed to optimize the temporal trace of cropping windows. Experimental results are presented not only by figure illustrations like the existing works did, but also by subjective evaluations with statistical analysis. Results shown that our content modeling algorithm is statistically significantly better than saliency toolbox, and our retargeting framework is at least similar to or slightly better than the back tracing. Last, we offer a series of adaptation demos at website: http://plaza.ufl.edu/lvtaoran/demo-all.htm. Fig. 2-16 provides a snapshot of our test videos.

Figure 2-1. Numerous digital video applications seek a smart content adaptation approach.

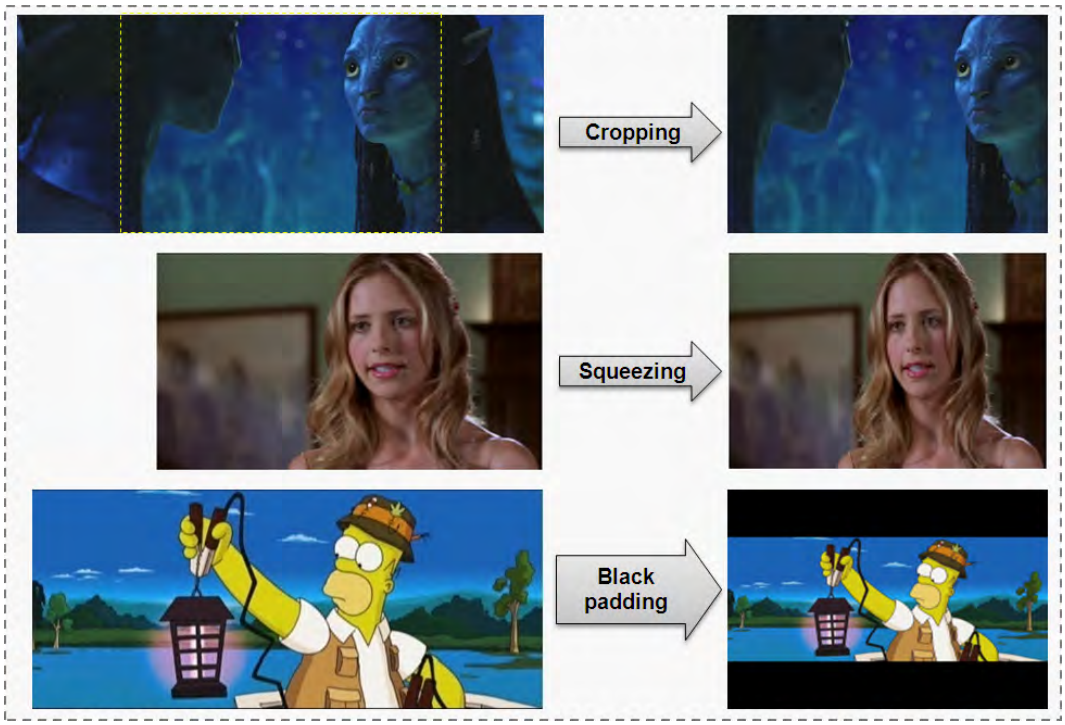


Figure 2-2. Examples of content-unaware video retargeting methods.

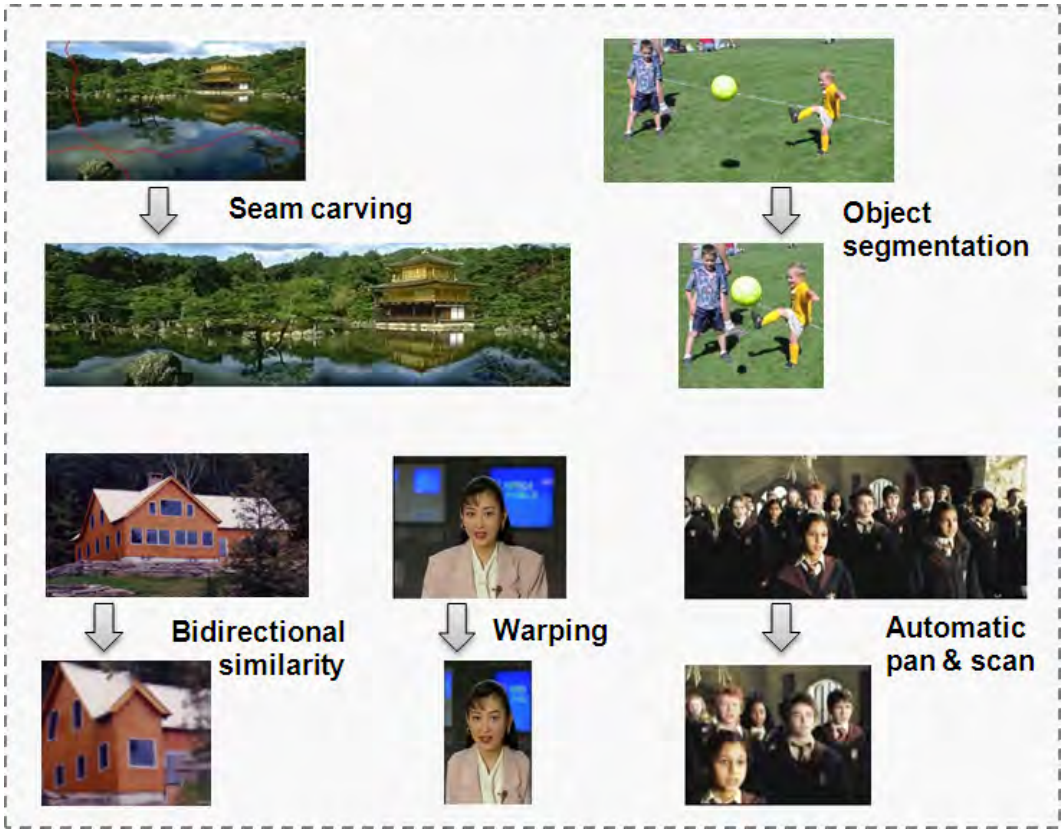


Figure 2-3. Examples of content-aware retargeting methods.

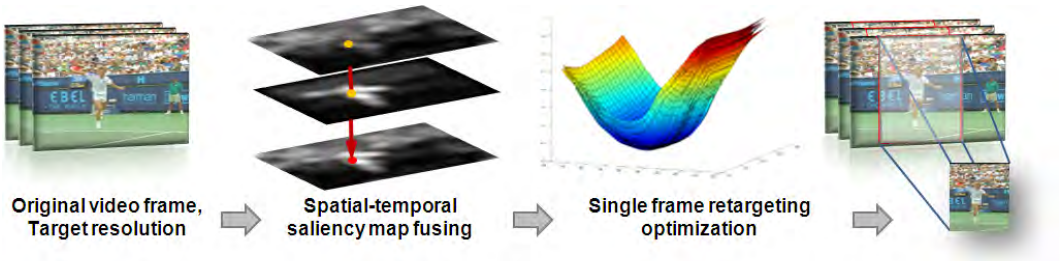


Figure 2-4. Overview of the nonlinear-fused spatial-temporal saliency detection and single-frame retargeting framework.

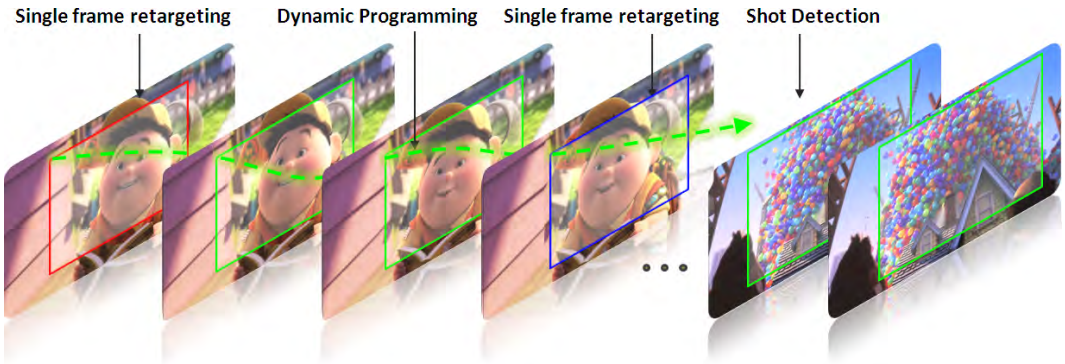


Figure 2-5. Overview of cropping window trace optimization framework on a shot/subshot basis, green arrow: search for optimized trace of cropping window throughout a subshot using dynamic programming.

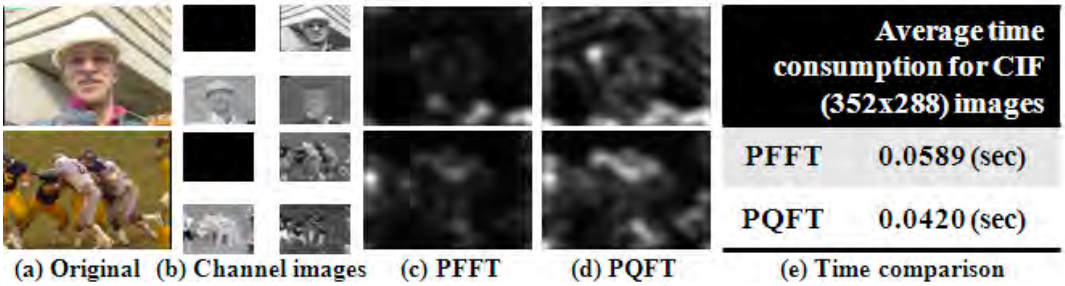


Figure 2-6. Comparison of spatial saliency detection by multi-channel PFFT vs. PQFT. a) Original frames, up: foreman, bottom: football. b) The $zero, Y, C_b, C_r$ channels. c) Saliency map detected by PFFT. d) Saliency map detected by PQFT. e) Time consumption.

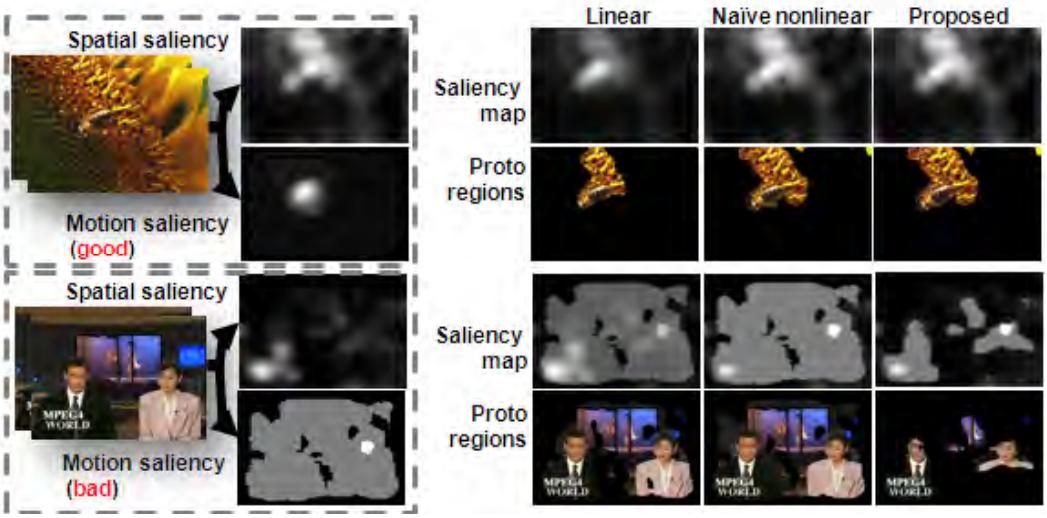


Figure 2-7. Comparison of linear combination, naive MAX operation and proposed approach when global motion is correct or wrong.

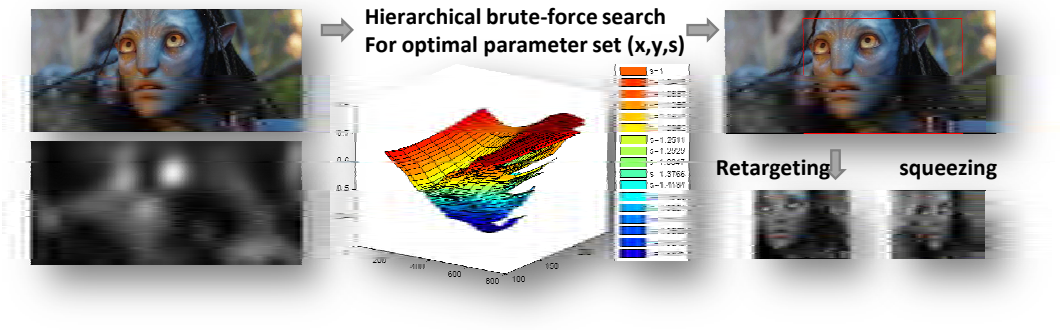


Figure 2-8. Left column up: a frame of AVATAR. bottom: spatial-temporal saliency. Middle column: Searching space of brute-force search. Right column up: cropping region on original frame. bottom left: retargeting result, bottom right: directly squeezing result.

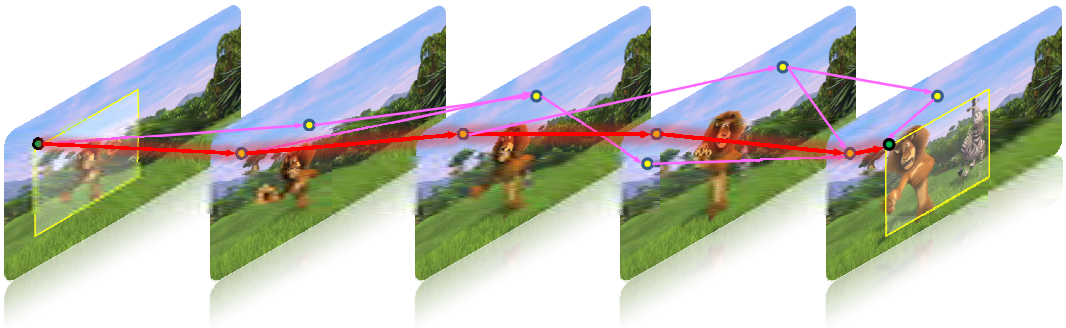


Figure 2-9. Graph Model for optimize cropping window trace; green: source and destination nodes. yellow: candidate node for each frame. red: shortest path to denote optimized dynamic trace.

Figure 2-10. Retargeting performances on natural images. Courtesy of [7] for the test images and results of comparison groups.

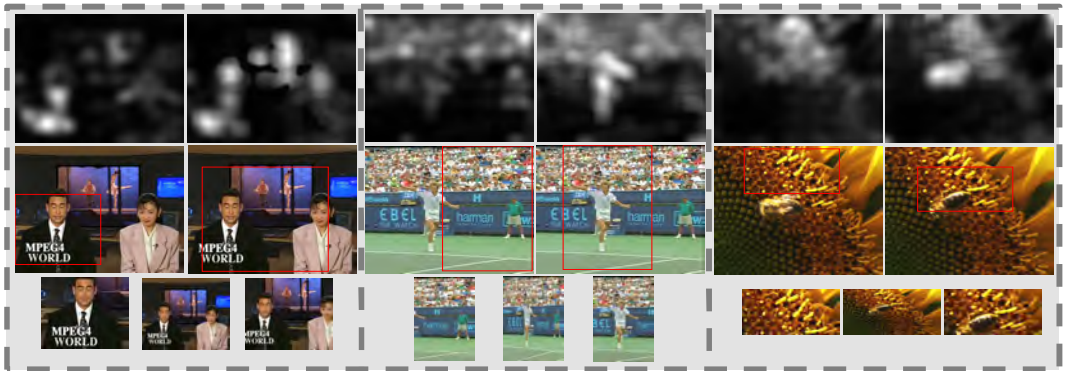


Figure 2-11. Comparison on video retargeting of baseline-PQFT [3] and our approach. For each video sequence, the left column shows results of baseline-PQFT, the right column shows ours. The first row are spatial-temporal saliency maps, second row are the optimal cropping windows and third row are the retargeting results. The middle figures in the third row are directly squeezing results.

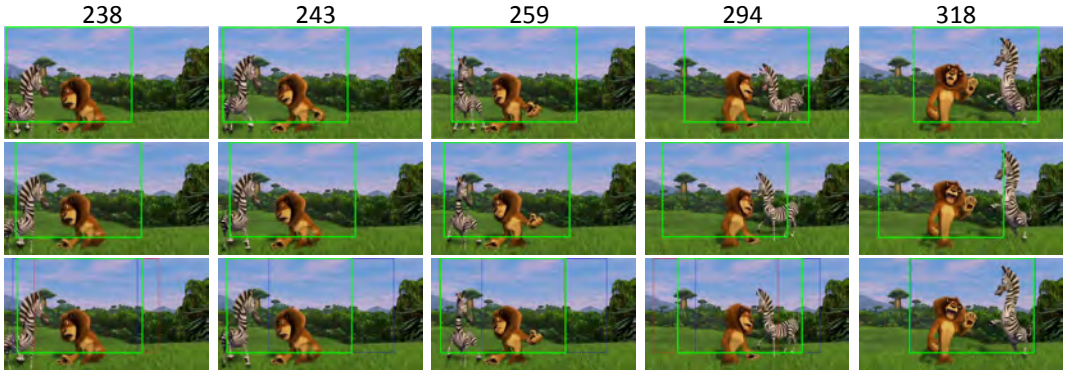


Figure 2-12. Retargeting Results: top: single frame search and smoothing, middle: back tracing, bottom: the proposed approach.

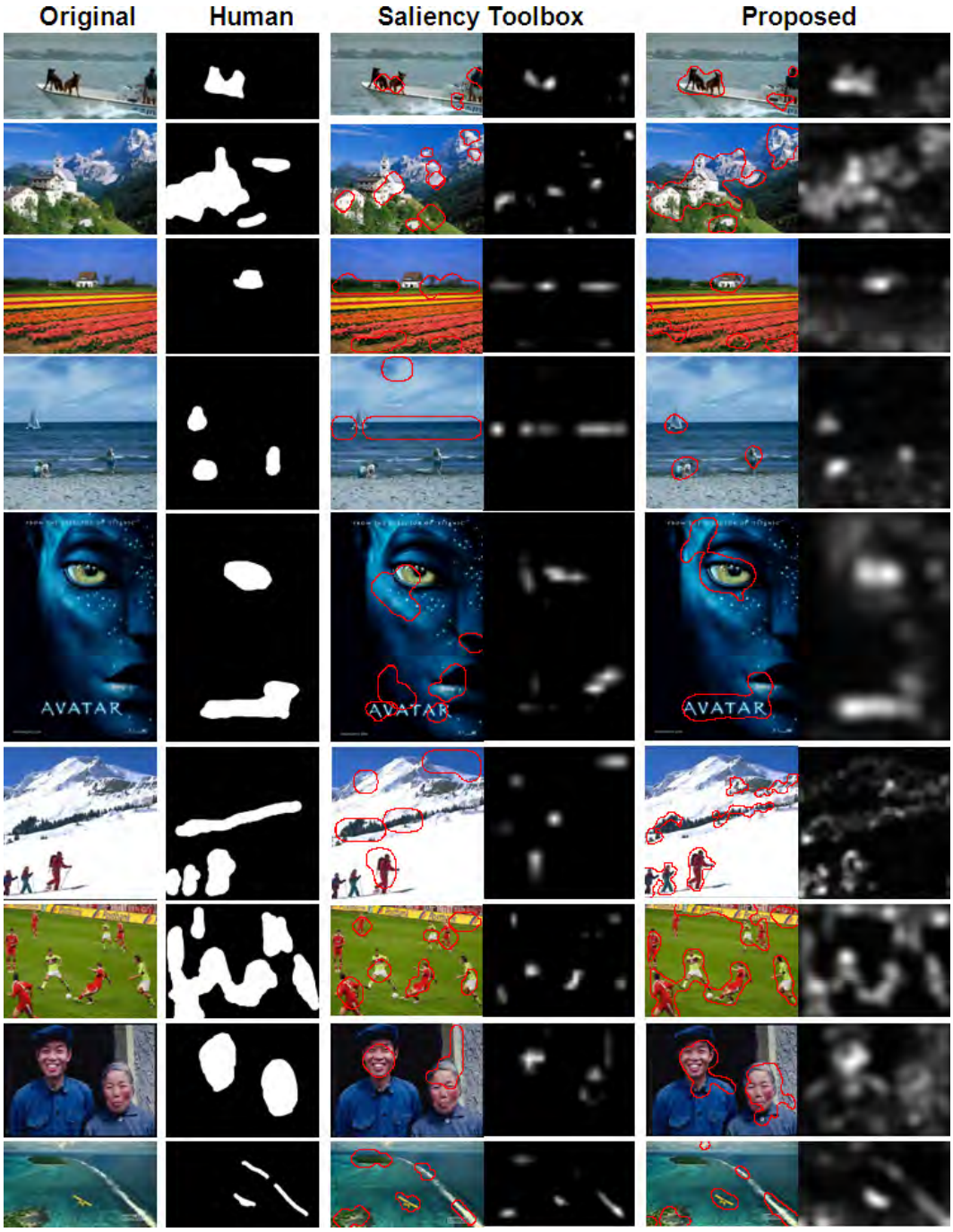


Figure 2-13. Comparison of saliency detection on images. Col.1: original image. Col.2: human labeled salient regions. Col.3: proto-regions detected by STB. Col.4: saliency map by STB. Col.5: proto-regions detected by our method. Col.6: saliency map of our method.

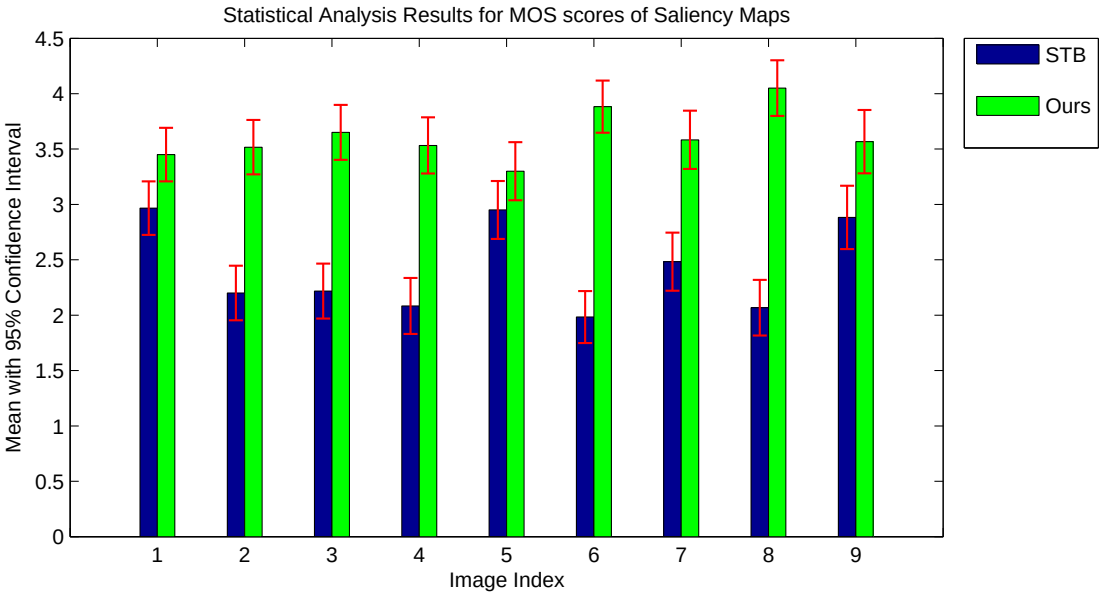


Figure 2-14. Statistical analysis for saliency maps. Blue: Saliency Toolbox. Green: Ours.

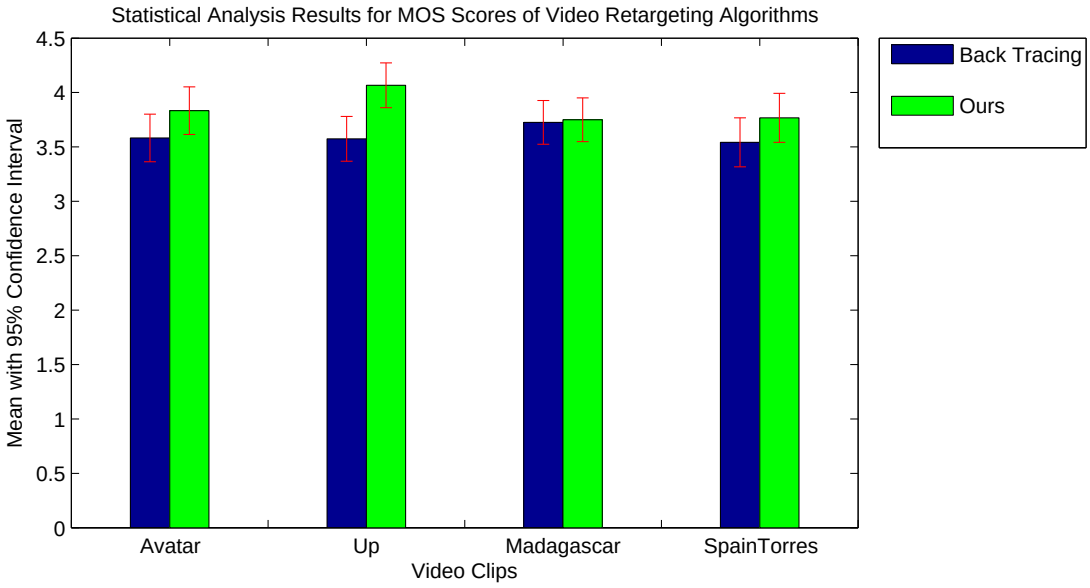


Figure 2-15. Statistical analysis for retargeting algorithms. Blue: Back tracing. Green: Ours

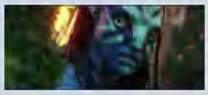
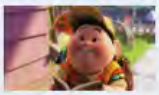
Name	Original Size	Target Size	Frame #	Preview
Avatar	848x360	300x200	534	
Up	852x480	320x240	513	
Madagascar	640x352	320x240	1560	
Godsend	1280x720	320x240	100	
FashionShow	640x360	320x240	1535	
SpainTorres	624x352	320x240	2064	
SpaceShuttle	640x360	320x240	510	
Coralreef	640x360	320x240	1589	
Sunflower	720x480	320x240	500	
Stefan	352x288	320x240	90	
News	352x288	220x176	300	
Jets	352x288	220x176	550	

Figure 2-16. A variety of test sequences.

CHAPTER 3
CONTENT-AWARE VIDEO SUMMARIZATION

3.1 Introduction

The fast development of digital video industry has brought many new applications and consequently, research and development of new technologies, which will lower the costs of video archiving, cataloging and indexing, as well as improve the efficiency, usability and accessibility of stored videos are greatly in demand. Among all hot research areas, one important topic is how to enable a quick browse of a large collection of video data and how to achieve efficient content access and representation. To address these issues, video summarization techniques have emerged and have been attracting more research interest in recent years.

There are two types of video summarization [17], static story board and video skimming (Fig. 3-1). Static story board [18] [19] [20] is a still abstract, which is a set of frames (key frames) selected from an original video sequence. Video skimming [21], also called a moving abstract, is a collection of image sequences along with the corresponding audio signals from an original video sequence. For the rest of this chapter, we will mainly discuss the moving abstract: video skimming.

The fundamental purpose of video skimming is to epitomize a long video into a succinct synopsis, which allows viewers to quickly grasp the general idea of original video. The resultant summary provides a compact representation of the original content structure, leading to efficient indexing and retrieval. Although brief, a good summary preserves all necessary hallmarks of the original video and viewers are sufficiently able to recover original content through reasoning and imagination.

3.1.1 Content-Unaware Video Skimming

An obsolete method of video skimming is to uniformly sample the frames [22] [23] [24] to shrink the video size while losing the audio part, like the fast forward function in digital players. Although this is probably the simplest way for video skimming, the drawback

is obvious. Such an approach may cause some short yet important segments to have no representative frames, while other longer segments may have multiple frames with similar content. This made the scheme fail to capture the actual dynamics of the video content. Time compression method [25] [26] [27] can compress audio and video at the same time to synchronize them, using frame dropping and audio sampling. However, the compression ratio of this method is limited unless the speech distortion is tolerable.

3.1.2 Previous Approaches of Content-Aware Video Skimming

To-date, there are excellent surveys on video skimming [17, 28]. These papers cover many detailed approaches with one common strategy: being formulated as an optimization problem, it selects a subset of video units (either static frames or dynamic shot clips) from all possible units in the original video such that they maximize some metric function of the summary quality.

Based on the cognitive level (from low to high: signal, syntax and semantic) where a metric function lies, we categorize current video skimming techniques into three types. Methods in Type I utilize signal level measures to compare the difference of a video summary from its original. Various implementations include the motion trajectory curve [29], visual redundancy [30], visual centroid [31], inter-frame mutual information [32], similarity graph [33] and summarized PSNR [34]. All these metrics are manipulations of pure frame intensities and in essence measure the visual diversity contained in a summary. Hence the maximization leads to the summary with most content diversity, deviating from the fundamental purpose of video summarization: It is a visually colorful one, but not necessarily the one that presents most important clues that enhance viewers' understanding.

Type II characterizes with high level semantic analysis, in which semantic events with explicit meanings are detected and the resultant semantic structure is utilized to guide the summarization. Generally, semantics are offline defined explicitly by some ontology, which annotates the events with meaningful tags. Through supervised

learning from labeled data, various methods in this category detects events with unique meanings. Typical implementations include the recognition of the “emotional dialogue and violent action” [35], “cinematography semantics” [36], “who and what inquires” [37], “lecture template” [38], “who, what, where and when entities” [39]. These methods make sense as they consider the fundamental purpose of video summarization closely. However, due to the limitation of current computer intelligence, recognizing an entity as an event with explicit meanings is a rigorous work as because of the well-known “semantic gap” problem. Also, the capacity limitations of a defined ontology force current approaches to exercise somewhat heuristic rules on the semantic entries, which prove to be powerful in ad hoc systems, but with weak generalization ability.

Type III lies in the intermediate level, with methods seeking entities with implicit meanings. The philosophy is that implicit semantical entities also suffice viewers to understand and recover original plot while avoiding the heuristic attempts for explicit semantic recognition. Some researchers in [40–43] assume the implicit semantics are expressed by popular human perception models and they yield summaries with most salient (most probable attended features by human attention) video units. Unfortunately, although correlated, salient features do not necessarily mean semantic distinguishable as they basically measure how interesting of a video while the interesting part may be an important clue for understanding or may be not.

3.1.3 Overview of the Proposed Approach

We feature our compelling video skimming algorithm to meet four requirements simultaneously. The four requisites are story skeleton preservation, appealing and salient summarization, smooth transition and skimming ratio adaptation. These demands impose huge challenges for us to come up with a well-formulated solution.

The first requirement is story skeleton preservation. Summary sequence enables viewers to quickly and efficiently grasp what a video describes or presents from a shorter summarized version. To meet this need, it is intuitive to extract the main skeleton

from the original video and keep it in the summarized video. Video skeleton could be seen as a queue of *concept primitives* with certain semantic implications in a temporal order. Concept primitive is not as high-level as real semantic concept, which is learned with human intervention. Here, the concept primitive implicitly encodes the semantic meanings of shots (sets of consecutive similar video frames), symbolizes shot group that portraits consistent semantic settings and generally possess the capability as a hallmark or self-evident clue that hints the development of the original video. Viewers may possibly recover the plot by only watching and hearing a handful of shots as long as all concept primitives are conveyed.

The second requirement is appealing and salient summarization. Obviously, an exciting summary of video is highly desired by viewers (i.e., highlights). Often, there are various shots conveying the same concept primitives. When selecting one shot conveying a concept primitive from many, the one with highest saliency value or equivalently generating the largest stimulus to human attention would be favored so that the resultant summarized video not only contains integral concept primitives, but also carefully selects shot instances with richest information to reflect these concept primitives to avoid a plain or even dull summarization.

The third requirement is smooth transition. An evident artifact in the summarized video is an unnatural transition between two adjacent concept primitives due to the elimination of a number of visually and acoustically similar shots. A compelling video summarization also expects smooth transition that requires frame level summarization besides of the concept primitive level.

Last but not least, the video skimming algorithm should have the scalability to adapt to arbitrary skimming ratio. For example, we may use different skimming ratios for one source video for different applications. It is highly desired that a video skimming algorithm can generate an informative and attractive sequence which satisfies both users' needs and the resource budget.

In this chapter, we propose a novel approach to explore the implicit semantics of original video on intermediate cognitive level. We pursue a self-explanatory video summary through discovering and preserving concept primitives. The motivation of concept primitive is intuitive: emulating the human cognitive process, naturally a list of key patterned hints, such as characters, settings, actions and their orders etc., are needed in the short summary for viewers to stitch these hints logically and use imagination to fill the omitted part. We extract audiovisual features and use spectral clustering to discover the concept primitives and consider the repetition of shot instances which instantiate the same concept primitive as summarization redundancy. We further analyze a good summary should keep various concept primitives as complete and balanced as possible so that the summary presents comparable clues from a complete perspective, allowing viewers to make most reasonable and objective inference. We also propose a greedy algorithm to apply the summarization criteria: based on the Concept-Primitive-Shot-Instance (CPRI) representation, we sort the shot instances primarily by the concept importance and secondarily by the saliency value within each individual concept primitive. Then shot instances are selected in a greedy fashion until the summarization ratio is reached. Finally, to meet the skimming ratio specification and keep the smooth transition in the summarized video, we add a frame level saliency thresholding followed by a temporally morphological operation as post processing. The main contributions of our work are therefore:

(1) A concept primitive based video summarization: its merits come from the fundamental purpose – to help viewers understand and recover the original plot semantically.

(2) Concept-primitive-shot-instance representation of video semantic structure: We proposed a unique way to discover the concept primitives using spectral clustering.

(3) A greedy approach to solve the summary problem with complete and balanced concept primitives as well as salient shot instances. This method is well suitable for scalability of the summarization.

3.2 Feature Extraction of Shots

3.2.1 Video-Audio De-Interleaving and Temporal Segmentation

A good video summary cannot be achieved without good understanding of the contents. The most common contents for a typical video sequence are visual and acoustic channels. Most of time, visual signals provide the majority information to learn latent concept patterns from original video; but audio sensory channels can also provide important information of a concept primitive in situations where visual channel may not offer, such as in an environment lack of light at nighttime. In addition, recall that a concept primitive implies that the comprising shots share both visual and audio consistency at the same time. Thus, if we allow independent feature extraction and unsupervised concept learning from both visual and audio sensory data, the learned concept results can be jointly analyzed in a parity-check fashion to enhance co-reliability. Therefore, we extract audio stream from raw video and put it through a parallel assembly line similar as visual stream does to discover the possible audio concepts.

The temporal segmentation for video stream is shot detection. We propose a variance-difference based approach to detect shot change, which is robust to detect the cut and can also get good performance to detect fades. The variance Var_i of frame i is calculated and the delta variance ΔVar_i of frame i with its previous frame $i-1$ is recoded. The algorithm for shot detection is presented as in Algorithm. 2.

For processing convenience, audio data are segmented into pieces, where each piece has its boundaries synchronized to its co-located video shot in time axis. Within each shot, audio data are further segmented into pieces with same time duration like a video frame. We call this kind of segments “audio frames”.

```
input  : Number of frames in sequence N, Thresholds T_stable, T_tolerance, T_length,
         Accumulated frame number in a shot N_s.
output: Shot index K_i.
K_0 ← 1, MaxΔ_Var ← −∞, MinΔ_Var ← ∞, N_s ← 0;
compute Var_0;
for i ← 1 to N − 1 do
    compute Var_i;
    ΔVar_i ← Var_i − Var_{i−1};
    if ΔVar_i < MinΔ_Var then
        MinΔ_Var ← ΔVar_i;
    end
    if ΔVar_i > MaxΔ_Var then
        MaxΔ_Var ← ΔVar_i;
    end
    if ΔVar_i < T_stable then
        if MaxΔ_Var − MinΔ_Var > T_tolerance then
            if N_s > T_length then
                New shot boundary found;
                K_i ← K_{i−1} + 1;
                reset parameters for a new shot;
                N_s ← 0;
                MaxΔ_Var ← −∞;
                MinΔ_Var ← ∞;
            end
        end
    end
    else
        N_s + +;
        K_i ← K_{i−1};
    end
end
```

Algorithm 2: Proposed algorithm for shot-detection.

3.2.2 Content-Aware Attention Modeling

An appealing summarization requires a content attentiveness (saliency) measurement. The saliency measure should effectively reflect how attractive a shot or a frame is. Based on our research on attention modeling in Chapter 2, we develop a saliency measure system.

There are three levels for visual saliency and two levels for audio saliency: pixel-level for visual saliency only, frame-level and shot-level for both visual and audio

saliency. Let SM_t be the spatial-temporal saliency map for frame t detected by our algorithm in Eq. 2–6. Then SM_t is the pixel-level visual saliency that indicate how attentive each pixel in the frame is. Frame-level visual saliency is measured as:

$$Sal_t^v = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} SM_t(i,j) \tag{3–1}$$

where (i,j) is the pixel location and W, H are frame width and height.

Frame-level audio saliency is measured by some low-level audio features [44], including spectral centroid (SC), root mean square (RMS), absolute value maximum (AVM), zero-crossing ratio (ZCR), and spectral flux (SF). SC is the center of the spectrum; it is computed considering the spectrum as a distribution which values are the frequencies and the probabilities to observe these are the normalized amplitude. RMS is a measure of short time energy of a signal from L_2 norm. AVM is a measure of short time energy of a signal from L_1 norm. ZCR is a measure of the number of time the signal value cross the zero axe. These low-level features can either be used alone or fused, then we obtained the frame-level audio saliency Sal_t^a. The frame-level audio-visual saliency is measured by a linear weighing with weighing factor λ:

$$Sal_t = \lambda Sal_t^v + (1-\lambda) Sal_t^a \tag{3–2}$$

For a shot, the visual and aural conspicuousness are calculated by averaging the frame-level saliency in that shot, respectively:

$$AvgSal_k^v = \{\frac{1}{N_k} \sum_t Sal_t^v | F_t \in Shot_k\} \tag{3–3}$$

$$AvgSal_k^a = \{\frac{1}{N_k} \sum_t Sal_t^a | F_t \in Shot_k\} \tag{3–4}$$

where N_k is the number of frames in $Shot_k$. The shot-level audio-visual saliency is measured by:

$$AvgSal_k = \lambda AvgSal_k^v + (1-\lambda) AvgSal_k^a \tag{3–5}$$

3.2.3 Concept Primitives and Bag-of-Words Features

Skeleton preservation requires some distinctive feature for shot-discrimination, the shot feature should be discriminative enough at the sense of representing the video skeleton, as it will be used to find the similarity among shots.

We propose to use the bag-of-words (BoW) model to characterize the shot properties in visual and aural domains, respectively. BoW model [45] was initially utilized in natural language processing to represent the feature of a text document. It considers each text document as a collection of certain words belonging to a reference dictionary but ignores the order and semantic implications of words. BoW model uses the occurrence of each word in the dictionary as the feature of text, thus it often ends up as a sparse vector. The BoW model can be regarded as the "histogram representation based on independent features". In our case, a shot can be regarded as a text document. However, since neither the "visual word" nor the "aural word" in a shot is ready for use like the real words in text documents, the "words" need to be well defined. It usually involves two steps to obtain a "word": feature extraction and codeword generation.

1) Visual concept primitives and Bag-of-words features

We interpret a video concept primitive as a self-learned set featured by a combination of certain *spatially local visual atom* (SLVA) and each SLVA stands for a single visual pattern, which is found within a localized neighborhood at a particular spatial location, with implicit semantic implications, like rose, butterfly, etc. A noticeable property of the video concept primitive is that, we only attach importance to the occurrence of SLVA's, without esteeming their order (spatial location). For example in Fig. 3-5, a shot of a far view of rose and butterfly and a close-up look of the same entities should both imply the same semantic implications, despite the rose and butterfly may appear in different locations and in different scales. The BoW model for visual shots, which graciously expresses the order-irrelevant property, is thus adopted, with the

SLVA's as the visual words. We adopt the scale-invariant feature transform (SIFT) feature extraction algorithm [46] to obtain the video words, because the SIFT feature best exhibits the local characteristics within a neighborhood, with highest matching accuracies under different scale, orientation, affine distortion, and partially invariant to illumination changes. Evaluations suggest strongly that SIFT-based descriptors are the most robust and distinctive, and are therefore best candidates for SLVA's.

Consider a regular full process mode, that SIFT feature points should be detected on every frame in the shot and on every region within a frame. This procedure, although precise, is especially time-consuming. Thus, some pre-processing need to be conducted before the SIFT feature detection We adopt key-frames to balance the computation cost and accuracy. Since frames within a shot appear to have minor differences, it is wise to select one frame as the most representative one, i.e. key-frame. There are many key-frame selection methods. Some straightforward methods include choosing the first / last frame, or the middle frame in a shot. Some motion-based approaches use motion intensity to guide the key-frame selection, like in MEPG-7 [47]. Unlike those approaches, we consider human attention models in the shot, and select the most salient frame to be the key-frame t_k:

$$t_k = \arg\max_t \{Sal_t^v \mid F_t \in Shot_k\} \tag{3–6}$$

The key-frame selection will save a huge amount of computation at minor cost of precision loss, under the assumption that the frames are similar within a shot. In addition, by exploiting the attention model on a single frame, we can further exclude some inattentive regions on the key-frame. We define active region AR_{t_k}on the key-frame by thresholding the saliency map:

$$AR_{t_k}(i,j) = \{F_{t_k}(i,j) | SM_{t_k}(i,j) > T, 1 \leq i \leq W, 1 \leq j \leq H\} \tag{3–7}$$

T is the active threshold. The SIFT feature detection on active regions will generate prominent and robust SLVA's of the frame. Fig. 3-2 illustrates the results of saliency masking on two shots from sequence "big buck bunny".

We adopt Lowe's algorithm [46] for SIFT feature detection in active regions on the key-frame. The frame is convolved with Gaussian filters at different scales, and then the differences of successive Gaussian-blurred versions are taken. Key points are located as maxima/minima of the difference of Gaussians (DoG) that occur at multiple scales. Then, low contrast key-points are discarded, high edge responses are eliminated. After that, each key-point is assigned one or more orientations based on the local gradient directions. Finally, a highly distinctive 128 dimension vector is generated as the point descriptor; i. e., the SLVA. Fig. 3-3 illustrates the SIFT feature detection results on two shots for sequence "big buck bunny".

After SIFT feature points are found on the key-frame of each shot, the shot as a bag has a collection of "visual-words", each one is a vector of dimension 128. The number of words is the number of SIFT feature points on the key-frame. A shot bag with its SIFT feature descriptors can now be regarded as a text document that has many words. In order to generate the histogram representation as the feature for the shot, "dictionary" should be built as the collection of all the "words" from all the bags, and similar "words" should be treat as one "codeword", like in text documents, "take", "takes", "taken" and "took" should be classified into one group and use "take" as the codeword for this group. A codeword can be considered as a representative of several similar SLVA's. We use K-means clustering over all the SLVA's, the number of the clusters is the codebook size (analogy to the number of different words in a text dictionary). Codewords are the centers of the clusters, and each "word" is mapped to a certain codeword through the clustering process. Thus, each shot can be represented by a histogram of the codewords. Fig. 3-4 shows the histogram-like representation of the BoW feature, and Fig. 3-6 shows the flowchart of visual BoW feature extraction for a shot.

2) Audio concept primitives and Bag-of-words features

Similar to the analysis for video channel, we explore the audio structure by audio concept primitives, rather than from more detailed single acoustic source level as in many audio recognition problems or even from further waveform-level perspective. In general, we interpret an audio concept primitive as acoustic environment featured by a combination of certain *temporally local acoustic atom* (TLAA) and each TLAA stands for a single audio pattern with plausible semantic implications (e.g. the audio concept conversation between John and Mary at the shore is featured as a combination of John's short time voice (a TLAA) switching with Mary's (a TLAA) and continuous environmental sound of sea wave (a TLAA)). Note that for the purpose of video summarization, we seek an audio skeleton that are usually comprised of "self-contained" concept primitives. By "self-contained", we mean that in the set of shots that form this concept primitive, every shot has TLAA's from the same closed subset of plausible audio patterns and the reshuffling of plausible audio patterns are allowed. This assumption originates from the fact that humans recognize an audio scene from a macroscopic perspective, which emphasizes the components instead of exact time and location of every component. Like in the example above, if another audio scene also include John, Mary and sea wave, but this time John continuously talk at the first half and Mary at the second half, without any voice switching. We still consider this scene is in the same concept primitive as the example above because it also convey the semantic implication that John Mary's conversation at the shore. So we assume in one audio concept, those shots are subject to consistent TLAA compositions, no matter in what order these TLAA's are arranged. In the context of audio concept clustering, at this level, the feature vectors of different shots may be much closer as long as their acoustic component TLAA's are alike. Then they are prune to be clustered into the same group, which captures the underlying common characteristics of an audio scene. comparing with many indicator-like features, which identifies a shot as a single acoustic source each shot will end up to be a sparse vector

with only one 1-entry that indicates which acoustic source this shot belongs to. This hard-decision-like feature is generally contradictory to the fact that an audio segment corresponding to a shot usually consists of multiple intervening sources., while this fact is implicitly reflected by BoW feature. For the indicator-like features, their sparse nature of shot data highlights the difference of shot data by assuming shot as a single source with majority contribution, which are usually different. In this way, the clustering may lose much opportunity to learn a reasonable concept primitive where shots have similar acoustic components but the majority sources are different.

To serve the need of concept primitive mining which focuses on the components rather than their order, the BoW model is quite suitable to represent the audio feature of a detected shot. If we chop the audio stream of a shot into multiple overlapped short-time audio segments with equal length, we may regard the shot as a bag containing multiple audio segments as audio words. Each word, with extracted feature by Matching Pursuit decomposition [48], represents a unique TLAA, which is audio pattern with plausible semantic implications; and a shot is consequently considered as a bag containing the audio patterns. The histogram of each word occurrence is a summarized feature of a shot through all the words within. Here, an encoding theme is applied to avoid the over-sparsity of feature vectors (negatively impact the classification result) from direct word occurrence statistic. We store all audio words from all shots in raw video into a dictionary, and conduct k-means clustering over the dictionary to produce k codewords. Then each word is assigned to a nearest codewords. The BoW feature of each shot is the occurrence of codewords inside.

In order to improve the robustness of an audio BoW feature, we also apply the saliency masking method to take account for those audio words above an acoustic saliency level, thus to avoid the negative effect on the BoW accuracy exerted by low salient audio words, due to its small value compared with noise. Fig. 3-7 shows an example of audio saliency masking, which is a thresholding on the audio saliency curve.

In terms of feature extraction for a word, we use matching pursuit method similar to [49] to decompose the audio segment corresponding to a word into a series of predefined waveform basis. Although many acoustic features such as Mel-frequency cepstral coefficients (MFCC), linear predictive cepstral coefficient (LPCC) for recognition purpose are available, they are only suitable for structured audio streams, such as music or speech. Matching pursuit (MP), however, is able to feature ambient sound and other unstructured sound, thus access much more information to enhance the awareness of a latent concept primitive. For an audio word as a short-time audio segment with a certain length that produces one single TLAA, its unique acoustic characteristic can be encoded by a set of base functions in a reference dictionary and corresponding correlation coefficients. Using MP, we enable an efficient sparse representation of the audio segment. Note that in MP, bases of a given dictionary are selected by maximizing the energy removed from the residual signal at each step; so the sparse representation of a decomposition resulting from MP is the most efficient in the sense that the reconstructed signal based on the selected basis takes up a larger percentage than any other decomposition method. Here we refer a Gabor dictionary with Gabor waveform basis for its promising reconstruction efficiency. Each particular Gabor waveforms are indexed by its scale, frequency and translation from origin. For a fixed number of iteration steps, MP selects a Gabor basis from the Gabor dictionary with the maximum similarity to the audio segment residual in terms of correlation coefficients. The Gabor function is defined by:

$$g_{s,\mu,\omega,\theta}(n) = \frac{K_{s,\mu,\omega,\theta}}{\sqrt{s}} e^{-\pi(n-\mu)^2/s^2} \cos[2\pi\omega(n-\mu)+\theta] \tag{3–8}$$

where s, μ, ω, and θ are scale, translation, frequency and initial phase respectively.

Note that the bases in Gabor dictionary are all in 256 point length. To encode a short time audio segment as a TLAA vector by MP decomposition, we make the length of the short-time audio segment as 256 point as well to neatly align with the Gabor base

input : Signal $f(t)$, Gabor dictionary D.
output: List of coefficients (a_n, g_{γ_n}).
$Rf_1 \leftarrow f(t)$;
while $\|Rf_n\| < T$ **do**
  find $g_{\gamma_n} \in D$ *with maximum inner product* $< Rf_n, g_{\gamma_n} >$;
  $a_n \leftarrow < Rf_n, g_{\gamma_n} >$;
  $Rf_{n+1} \leftarrow Rf_n - a_n g_{\gamma_n}$;
  $n \leftarrow n + 1$;
end

Algorithm 3: Matching pursuit algorithm.

function. Applying MP, a TLAA can be represented by a feature vector each entry of which symbolizes the coefficients of a selected Gabor basis. The flow chart of audio BoW feature extraction is shown in Fig. 3-8.

3.3 Video Skimming by Concept Reconstruction

3.3.1 Spectral-Clustering Solution for Concept Learning

In the following parts of this chapter, we'll use concept to abbreviate the "concept primitive". With the feature vector available for each shot under both visual and audio sensory channels, shots in original video are ready for clustering to discover the latent concepts. Visual and audio sensory channels are processed independently so that they can provide mutual reliability to each other. A compelling clustering method would first be able to group the data correctly, even though the numbers of data in different clusters are considerably different. We incorporate spectral clustering [50] method to learn the possible concept from shots. Given shot feature data, spectral clustering provides a state-of-the-art classification approach. Spectral clustering minimizes an objective function that cancels out the negative effect due to imbalanced division of number of members in different clusters. Thus even though original video contain concept patterns that consist of significantly different number of shot members, spectral clustering is free of artificial bias of a division of uniform number of members and is capable of dividing them correctly as long as the feature measure make the shots in same concept consistent. Another benefit of spectral clustering is that it favors to

classify locally-correlated data into one cluster Because it adds another constraint to distinguish the close-located or locally-connected data and increase their similarity to be divided into one group. By this constraint, the clustering result approaches human intuition that a cluster with consistent members is generally subject to a concentrated distribution. By the virtue of spectral clustering, the latent concepts are independent from the number allocation of shot members in different clusters; meanwhile, due to the favor of locally-connected data into a single cluster, the learned concept tends to be self-contained, which is desirable to represent a video skeleton. The algorithm of spectral clustering is referred to as Algorithm. 4.

input : feature vector set $U = u_1, ..., u_n$, search range $[k_{min}, k_{max}]$.
output: clusters c_j.
form an affinity matrix A*:*
$d(u_i, u_j) = \|u_i - u_j\|$;
$A_{ij} = exp(-d(u_i, u_j)^2/2\sigma^2)$ if $i \neq j$ and $A_{ii} = 0$;
define D *to be a diagonal matrix whose* (i, i) *element is the sum of* A*'s* i*-th row*;
construct normalized affinity matrix L;
$L = D^{-1/2}AD^{-1/2}$;
eigen-analysis for L*: get* $k_{max} + 1$ *largest eigen-vectors* $(x_1, ..., x_{k_{max}+1})$ *and corresponding eigen-values* $(\lambda_1, ..., \lambda_{k_{max}+1})$;
estimate optimal cluster number:
$k = \arg max_{i \in [k_{min}, k_{max}]}(1 - \lambda_{i+1}/\lambda_i)$;
form matrix $X = [x_1 x_2 ... x_k] \in \mathbf{R}^{n \times k}$ *by stacking the first* k *eigenvectors in columns.*;
form matrix Y *by renormalizing each of* X*'s rows:*
$Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$;
cluster each row of Y *into* k *clusters via cosine-distance based K-means:*
assign original data point u_i *to cluster* c_j *if row* i *of* Y *is clustered to* c_j;

Algorithm 4: Spectral clustering algorithm.

Here the feature vector set U is our extracted feature set of visual-BoWs and audio-BoWs, respectively. The number of clusters k are the number of concepts, which can either be empirically set or adaptively adjusted by iteratively try different number of k, since the computation complexity of spectral clustering algorithm is fairly low

compared to other clustering algorithms like meanshift or K-means on original high dimensional data.

3.3.2 Audio-Visual Concept Alignment and Consistence Checking

After spectral clustering for both visual and audio sensory channels, the shots are grouped into different visual and audio concepts. Note that for a scene with both aural and visual contents that conveys a certain concept, its visual layer and audio layer are only physically different carriers, however, they present the same semantic implications for the same concept. Therefore, we assume a one-to-one mapping from the learned visual concepts to audio concepts. We propose a number-of-member based method to align visual and audio concept indexes. Since a visual concept with its audio counterpart reflects a single concept with semantic implications, the number of shot members in this concept would reveal the identity of the concept. Since the label index for audio and video clustering results are independently generated and the label index are randomly assigned, for example, “2 1 3 1 1 3” for video should represent same concept clustering result as “3 2 1 2 2 1” for audio. Thus we need to align the labels for an easy checking for consistency, i. e., rearrange the labels for audio to be “2 1 3 1 1 3”. This number-of-member is feasible also because spectral clustering imposes no such artificial effects to evenly divide data to clusters. The bimodal concept alignment algorithm is as Algorithm. 5.

When the audio and visual concepts are aligned, we should check if the concepts are consistent. Consider that some shots have mismatched audio-visual concepts, for example, a video of two people A and B talking; most shots will consistently show the person’s figure and play the person’s voice. Some shots will show A’s figure while play B’s voice. The case is rare but possible, and we call it a mismatch. After the concept alignment, the mismatch flag d_k for shot $k = 1...K$ can be easily found by comparing the aligned spectral clustering results:

$$d_k = 1, \text{ if } V_k \neq A_k, \text{ else } d_k = 0 \tag{3–9}$$

```
input  : Visual concept label V_k and audio concept label A_k for shot k, k = 1...K.
output: Aligned concept labels.
VM_l ← 0, AM_l ← 0;
for k ← 1 to K do
    for l ← 1 to N_cluster do
        if V_k = l then
            VM_l ← VM_l + 1;
        end
        if A_k = l then
            AM_l ← VM_l + 1;
        end
    end
end
sort VM and AM in descending order and get their index mapping I.
VM_l ← 0, AM_l ← 0;
for k ← 1 to K do
    for l ← 1 to N_cluster do
        if A_k = l then
            A_k ← I_l;
        end
    end
end
```

Algorithm 5: Proposed audio-visual concept alignment algorithm.

When there is a mismatch, the audio-visual saliency of the shot should be decreased, since keeping such kind of a shot in the skimmed video will cause some misunderstanding to viewers.

$$AvgSal_k = AvgSal_k - \alpha d_k \tag{3–10}$$

where α is saliency penalty for audio-visual concepts mismatch.

3.3.3 Skimming Algorithm and Post Processing

We propose a greedy algorithm to progressively generate the summarized video clip, by the means of collecting shots. In other words, a video skimming process can be regarded as a video reconstruction process; starting from an empty output sequence, a shot is recruited each time to the output sequence, until the target skimming ratio is achieved. The duration of the output video can thus be controlled by recruiting

different amounts of video shots to satisfy arbitrary skimming ratio. A crucial factor is the recruiting order, which plays an important role to the final result. Given the requirements, we design several rules and propose a *reconstruction reference tree* structure for our skimming algorithm:

Rule 1: Concept integrity should be satisfied. The concept integrity, or concept completeness, is the major concern of our skimming scheme. We regard the ultimate goal of video skimming is to faithfully reflect the diversity of concepts of the original video thus yield the maximum entropy, despite that some concepts may seem not salient (exciting). Therefore, in our reconstruction framework, we require that each concept should contribute shots to the skimmed video.

Rule 2: Concept importance should be measured. The concept importance is a factor for deciding the recruiting order of different concept primitives. It is not equivalent to the concept saliency. It is a more high-level argument that will reveal the video producer's intention for the concepts' representation. Most commonly, if the producer gives a long shot for a concept primitive, or repeats the concept in many shots, then this concept is of high importance intentionally. Under this assumption, we can assign the concept importance Im for concept C_l as:

$$Im_l = \{\sum N_k | Shot_k \in C_l\} \tag{3–11}$$

where N_k is the number of frames in shot k. In our reconstruction framework, we require that a shot should be first pick from the most important concept.

Rule 3: The overall saliency value of the output sequence should be maximized. As the concept diversity is guaranteed by Rule 1, we can now focus on the saliency requirement – maximizing the saliency. For similar shots in one concept cluster, the shot with highest saliency should be pick up first. We define several terms for better describe the rules in our algorithm.

1) *Must-in shot* and *optional shot*

We define the most salient shot in each concept a “must-in” shot. It means that the shots must be recruited in the skimmed video regardless of the skimming ratio. This guarantees the concept integrity. The other shots are “optional” shots that can be recruited or not depending on the target skimming ratio.

2) *Reconstruction reference tree*

The reconstruction reference tree (RRT) is a data structure we designed for video reconstruction guidance. It is built according to the rules defined above. The root of the RRT is the video concept space, which is the set of learned visual concepts through the spectral clustering process. The first level leaves are the concepts, sorted in importance descending order from left to right. The second level leaves are the shots. Under each concept, the shots are sorted in saliency descending order from top to bottom. The first child of each concept is the must-in shot and the rests are optional shots.

3) *Virtual shot* and *shot table*

Since each concept may have different number of shots, we put some “virtual shots” with zero saliency to form an array of all shots. The array is called the shot table. The must-in and optional shots are *real shots*.

An RRT with must-in shots, optional shots, virtual shots and shot table is illustrated in Fig. 3-9. Given the RRT and shot table, the reconstruction process is relatively easy. The algorithm operates iteratively, for each time, a real shot is picked up from the shot table, with raster scan order, until the actual skimming ratio exceeds the target skimming ratio. Since the reconstruction is based on shots, the actual skimming ratio R_{act} may not perfectly equal to the target skimming ratio. It is more likely that R_{act} is slightly larger than R_{tar}, as the the stop criteria is that R_{act} exceeds R_{tar}. In order to precisely control the output video duration, we propose to use pure frame-level skimming, which is based on the attention model, as post processing. The audio-visual saliency defined previously of every frame that appears in the output sequence is checked again; through thresholding on the saliency curve, the frames with relatively

low saliency will be discarded, making the final duration of the output video satisfy the target duration. In addition, the smoothness requirement is also considered to yield a viewer-friendly skimmed video. A morphological-like operation [40] is adopted (denoted as function morph()): delete curve segments that are shorter than K frames, and join-together curve segments that are less than K frames apart, where K is a small value empirically set. Algorithm. 6 describes the video skimming and post processing process. Fig. 3-10 illustrates an example of saliency curve thresholding with curve preserving ratio $R = 95\%$.

3.4 Experimental Results

As content-aware video skimming is a highly subjective task like content-aware video retargeting, it is also difficult for any mechanical comparison or simulation methods to obtain accurate objective evaluations, and there does not exist a standard method to evaluate or quantify the performance. Thus, we also present our results with both figure illustrations and subjective tests, as we did previously for content-aware video adaptation.

3.4.1 Figure Illustrations

We first examine the capability our algorithm to mine the video concept primitives in a video sequence, by doing shot detection and extracting shot features. Since a very long and complex video may contain too many concept primitives, it is very likely that different human beings yield different clustering results of shot groups. To avoid the ambiguity of the ground truth, we illustrate the results for concept mining using a 30 seconds clip, which has a clear concept primitive, from the popular sitcom “the big bang theory”. This sequence shows a typical conversation of four people (Leonard, Sheldon, Howard, Rajesh) in a living room. We analyze this clip manually to generate the ground truth. The clip contains 17 shots, 5 concept primitives: Leonard talking (L), Shelton talking (S), Howard talking (H), Rajesh talking (R) and all people together (A). The story is progressively evolved as “A H L H R H L R A R A R S A R A S”. We employ

```
input  : Reconstruction Reference Tree: Concepts C_l, l = 1 : L and Shot Table
         S_{k,l}, k = 1 : K_max, Target skimming ratio R_tar.
output: Final skimmed video V_o.
V_o ← ∅, R_act ← 0, flag_must ← 0;
while (R_act < R_tar or flag_must = 0) do
    if S_{k,l} is a must-in shot then
        pick up this shot:
        V_o ← V_o ⊎ S_{k,l};
        update R_act;
        if l = L then
            flag_must = 1;
            k ← k + 1;
        end
    end
    if S_{k,l} is an optional shot then
        pick up this shot:
        V_o ← V_o ⊎ S_{k,l};
        update R_act;
        if l = L then
            k ← k + 1;
        end
    end
    if S_{k,l} is a virtual shot then
        skip this shot;
        if l = L then
            k ← k + 1;
        end
    end
    l ← l + 1;
end
// post processing
calculate frame level saliency curve SC for V_o where frame index is t;
calculate curve preserving ratio R ← R_tar/R_act;
SC ←MedianFilter(SC);
calculate threshold T such that R% of SC is above T;
SC ← {SC | SC_t > T};
SC ←morph(SC);
update V_o by collcecting frames t from SC;
```

Algorithm 6: Proposed algorithm for video skimming by shot reconstruction and post processing.

our algorithm 2 to do shot detection and key-frame selection using Eq. 3–6. Then, the novel saliency masking technique is applied on both visual and audio channels, and BoW feature for each shot are extracted by saliency-masked SIFT-feature detection on visual frames and MP decomposition on audio segments.

The first figure illustration, Fig. 3-11, shows the detected 17 shots of “the big bung theory” by their saliency-masked key-frames. It is shown that our shot detection algorithm successfully detected the 17 shots in this clip. The upper part of Fig. 3-11 shows the key-frames for each of the 17 shots. The key-frames are selected as the most salient frames in each shot. The black regions of each frame are the masked regions by saliency masking, a novel approach that we used to eliminate backgrounds and leave the salient regions for robust feature detection. The bottom part of Fig. 3-11 depicts the SIFT feature points detected on salient regions. We could see that similar shots have similar SIFT features detected. Thus are very robust for shot feature clustering in the following procedure.

The second figure illustration is Fig. 3-12. It shows the spectral clustering results for the 17 shots with the proposed BoW features. Five concept primitives are mined by the proposed approach. Each concept primitive consists of similar shots. For example, concept primitive $C2$ contains shot 1, 3 and 5 which show Howard talking (H). The clustering result matches with the ground truth perfectly for this sequence. Our shot clustering algorithm is shown to be effective for concept mining.

The third illustration, Fig. 3-13, presents the reconstruction reference tree (RRT) built for “the big bung theory”. The five concept primitives are arranged from left to right with decreasing concept importance. The shots belong to each concept pattern are arranged from top to bottom under each concept, with decreasing saliency. We can simply read from the RRT the order for picking shots to generate an output by reconstruction: 0 11 5 12 2 are the must-in shots that will definitely appear in the output sequence. Following are shots 15 14 3 16 6 13 4 1 10 7 8 9 which are optional shots

that their appearances in the output depend on the given skimming ratio. The skimming process is invoked with Algorithm. 6. Finally, the output sequence is generated.

3.4.2 Subjective Evaluation

To qualitatively measure how well our algorithm can generate a skimmed video, we employ the user study which has been widely used for video summarization evaluation [42] [40] [51] [52] [43]. We adopt two metrics *informativeness* and *employability* proposed in [42] to quantify the quality of the skimmed video under different skimming ratios. Enjoyablity reflects user's satisfactory of his/her viewing experience. Informativeness measures the amount of information of the original video that the skimmed video can preserve.

The subjective test is set up as follows. First, considering not to cause tiredness to participants to downgrade their viewing experiences, we carefully pick two testing videos. The first is a four minutes clip in "the big buck bunny" (BBB), from www.bigbuckbunny.org, and the other is a seven minutes clip in "lord of the rings" (LoR) from the MUSCLE movie database [53]. Then, we assign two skimming levels for each clip, 20% and 10% for BBB to test the extreme skimming cases and 50% and 30% for LoR to test ordinary skimming cases. Thus, two skimmed videos are generated for each of the testing videos. Next, we provide all the six video clips on our subjective testing website (http://www.mcn.ece.ufl.edu/public/subjective/), and give instructions on how to evaluate the output by the aforementioned metrics enjoyability and informativeness. The participants are then asked to give each skimmed video an enjoability score and an imformativeness score in percentage ranging from $0\% - 100\%$, to express their feelings on the skimmed videos. During the whole evaluation process, the participants could take their time to play each video as much time as they want, before they feel that the scores truthfully reflect their feelings.

We collect scores submitted by 60 participants after the subjective test. We then plot the histograms of the scores in Fig. 3-14. As is shown in the histograms, the

enjoyability and informativeness scores of 20% skimming to BBB exhibit peaks within the $50\% - 80\%$ range, which is a very promising result. The scores of 10% skimming to BBB are more uniformly distributed than the 20% scores, showing that participants' feelings diverse when the ratio approaches the extreme. For the histograms of scores to LoR with 50% and 30% ratio, the scores are more concentrated than the histograms of very low skimming ratios.

We calculate some basic statistics of the scores and present the statistics in Table 3-1 and Table 3-2. Accordingly, bar charts with confidence intervals are plotted to illustrate the data from the tables, as are shown in Fig. 3-15. The bars show the mean of enjoyability and informativeness scores with 95% confidence level for the two skimmed videos of BBB and LoR, respectively. It is clearly seen from the plot that the scores are significantly higher than the corresponding skimming ratio, which indicated that users agree that our skimming algorithm effectively generates an output video which exploits enjoyability and informativeness.

Table 3-1. Basic statistics of subjective testing scores of "big buck bunny"

Big buck bunny		Mean	Std. error	Std. deviation	Sample variance	Min	Max	95% confidence
20%	Enjoyability	61.25	2.82	21.86	477.65	10	90	5.65
	Informativeness	68.00	2.18	16.9	285.76	20	100	4.37
10%	Enjoyability	55.83	2.14	21.26	451.84	10	100	5.49
	Informativeness	58.95	2.17	16.78	281.40	10	90	4.33

Table 3-2. Basic statistics of subjective testing scores of "lord of the ring"

Lord of the ring		Mean	Std. error	Std. deviation	Sample variance	Min	Max	95% confidence
50%	Enjoyability	64.05	2.35	18.21	331.78	10	100	4.71
	Informativeness	66.08	2.42	18.71	350.07	20	100	4.83
30%	Enjoyability	57.25	2.56	19.82	392.73	10	100	5.12
	Informativeness	60.92	2.26	17.5	306.35	20	90	4.52

3.5 Summary

We have presented a novel approach for video skimming in this chapter. Audio-visual bag-of-words shot model and spectral clustering are incorporated for video concept

mining. Saliency-masked SIFT feature descriptors on key-frames are taken as visual BoW features and matching-pursuit decomposition is adopted to discover audio BoW features for a shot. Spectral clustering is employed for unsupervised concept primitive learning. Then, from the reconstruction perspective, the skimmed video is progressively generated in a greedy fashion with the reconstruction reference tree which takes into account both video informativeness and enjoyability under a given skimming ratio. Smoothness requirement is achieved by postprocessing. Our approach is shown by subjective tests to have encouraging results to offer both informative and enjoyable summarizations. Finally, we provide some content-aware video summarization demos at website: http://plaza.ufl.edu/lvtaoran/skimming.htm.

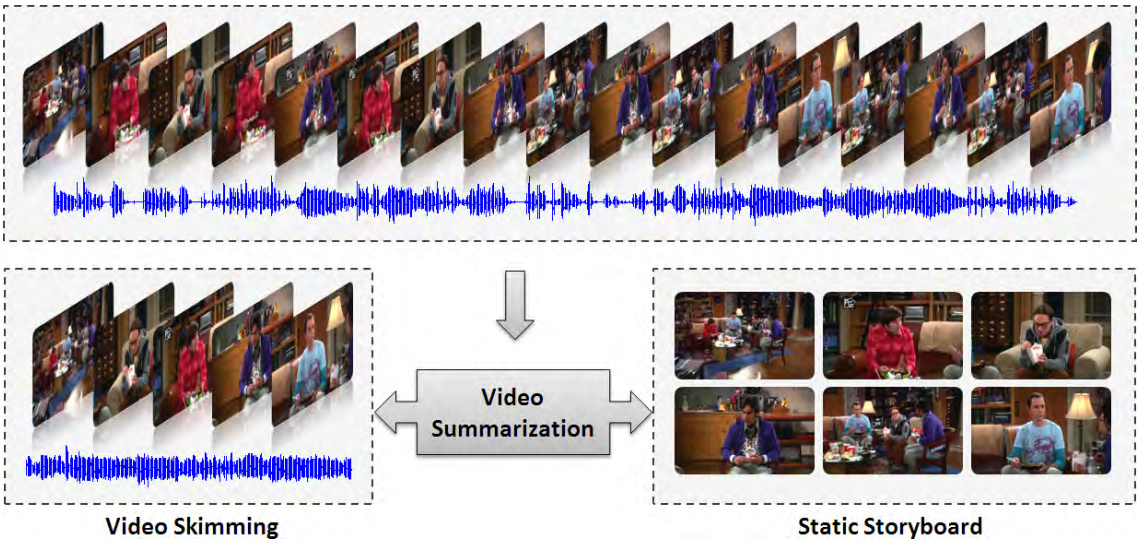


Figure 3-1. Video summarization techniques: static story board and video skimming.

Figure 3-2. Visual saliency masking on “Big-Buck-Bunny”.

Figure 3-3. SIFT feature detection on active regions. The arrow length for every point is the L_2 norm of the SIFT descriptor.

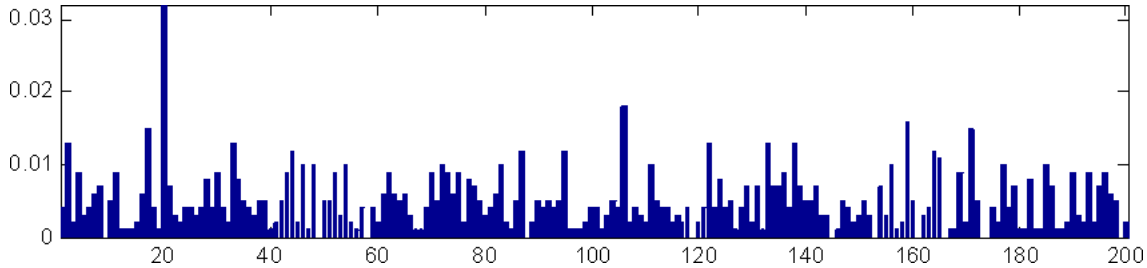


Figure 3-4. The histogram representation of the visual BoW feature for a shot.

Figure 3-5. Same semantics concepts in different scales and locations.

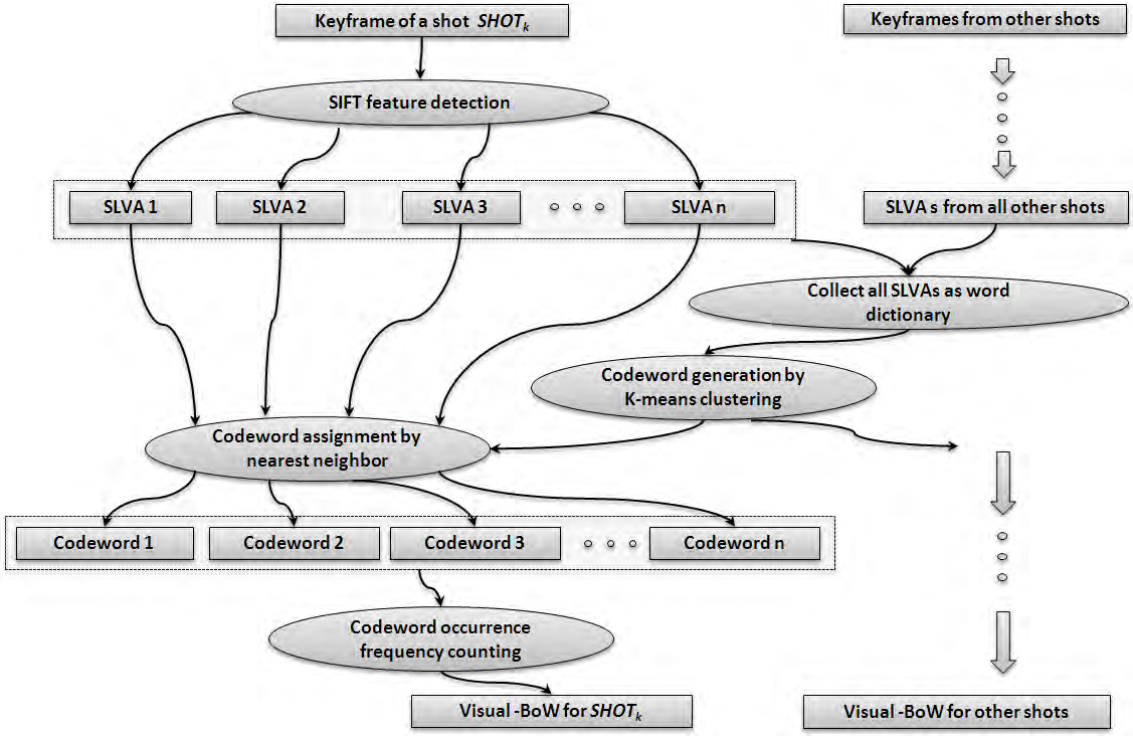


Figure 3-6. The flowchat for extracting visual BoW feature.

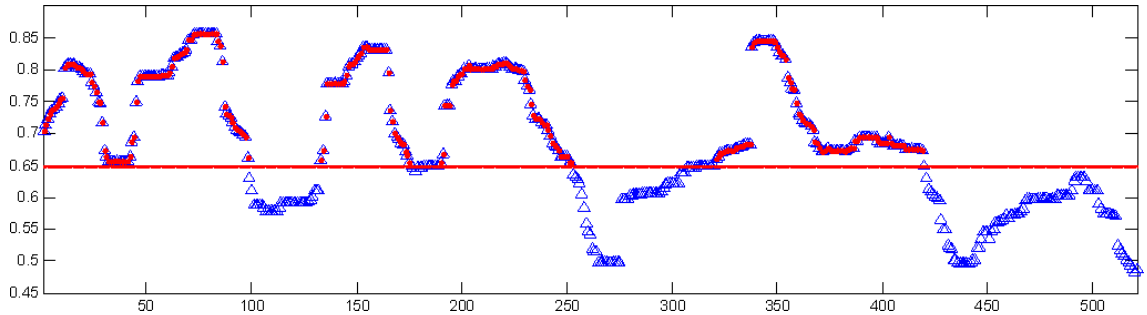


Figure 3-7. Audio saliency masking.

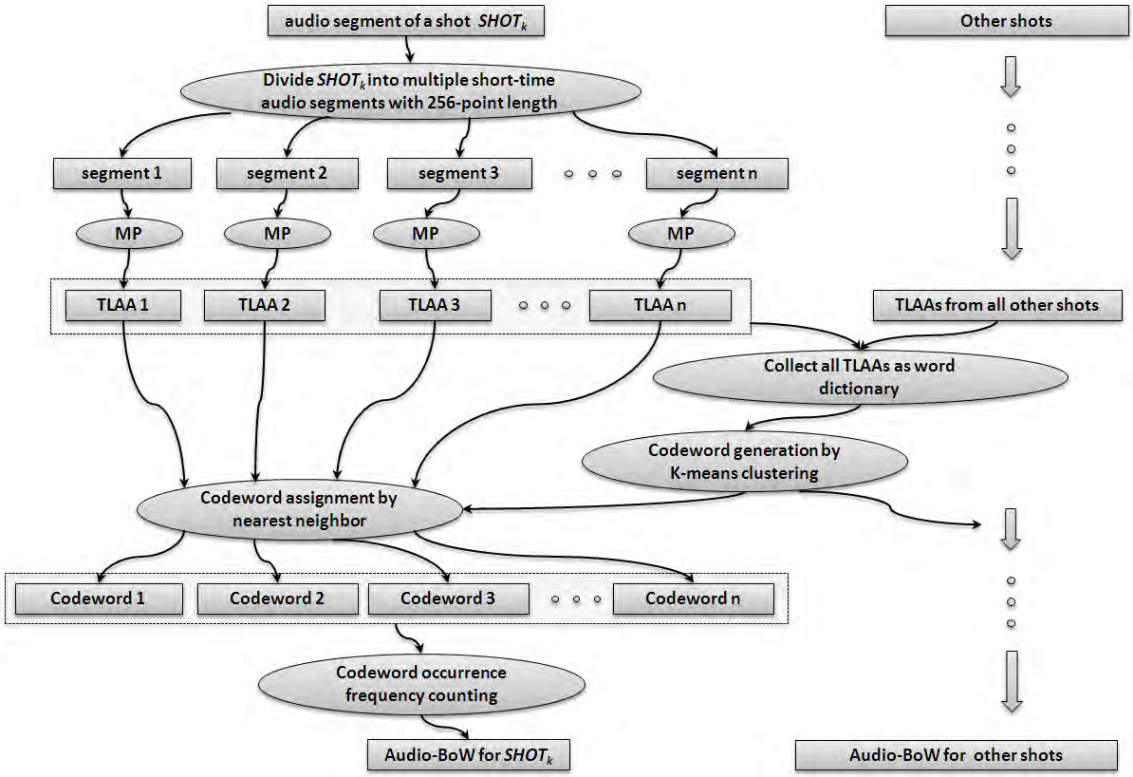


Figure 3-8. The flowchat for extracting audio BoW feature.

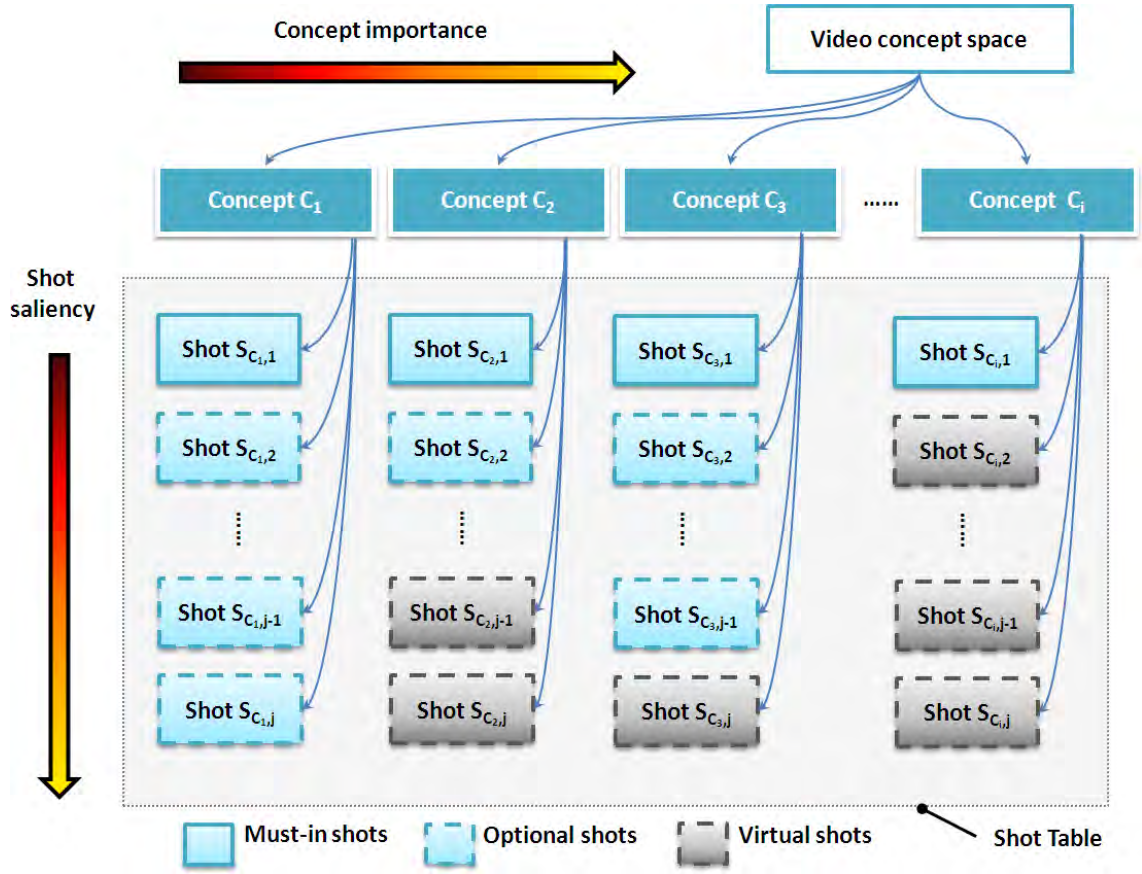


Figure 3-9. An RRT with must-in shots, optional-shots, virtual shots and shot table.

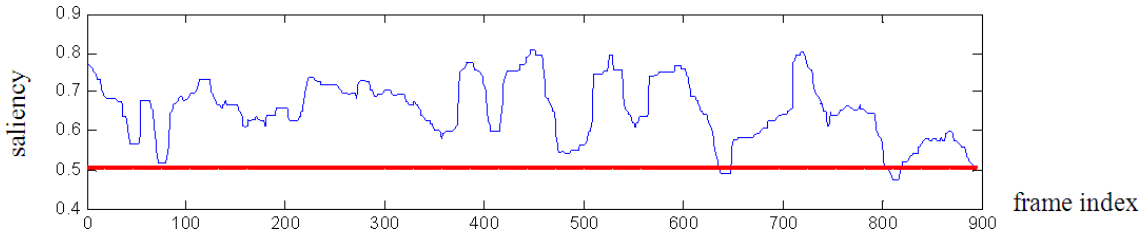


Figure 3-10. Post processing by saliency thresholding.

Figure 3-11. Up: The saliency-masking on detected 17 shots. Bottom: The SIFT features detected on key-frames of “The Big Bang Theory”.

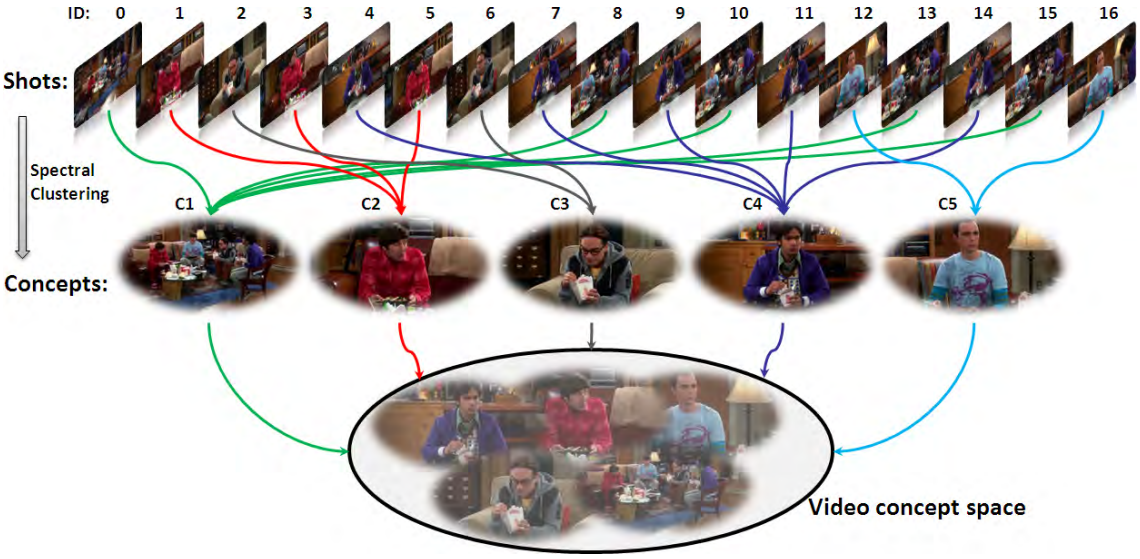


Figure 3-12. The concepts mining by spectral clustering of bag-of-word shot features of sequence “the big bung theory”.

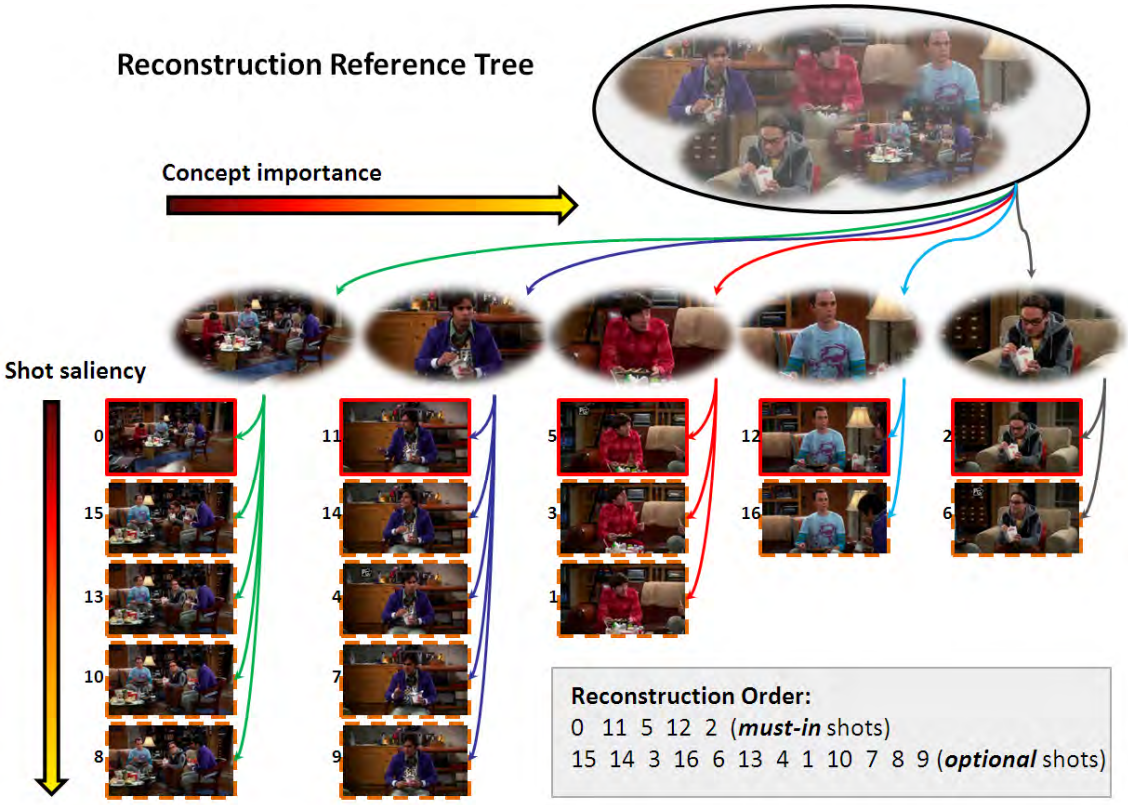


Figure 3-13. Reconstruction reference tree of “the big bung theory”.

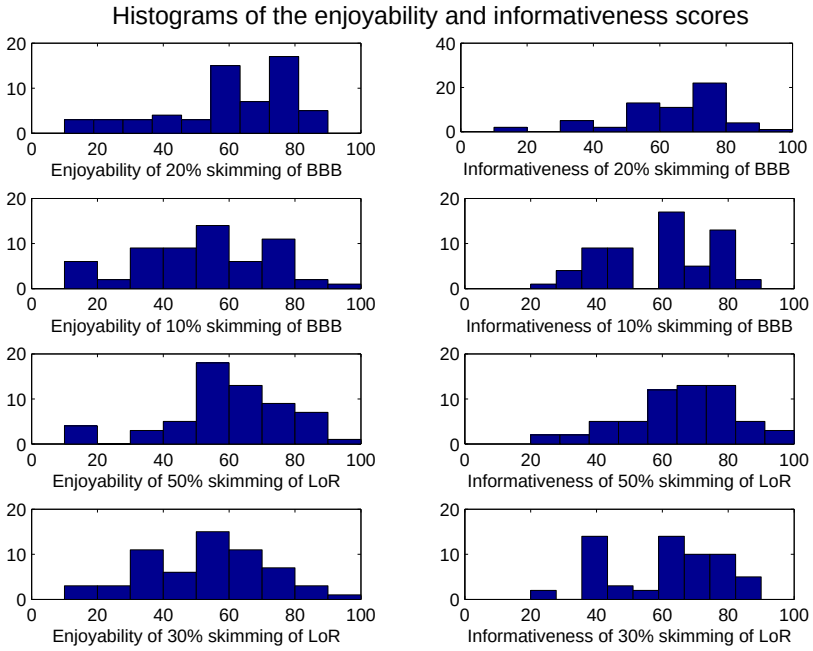


Figure 3-14. The histograms of the enjoyability and informativeness scores.

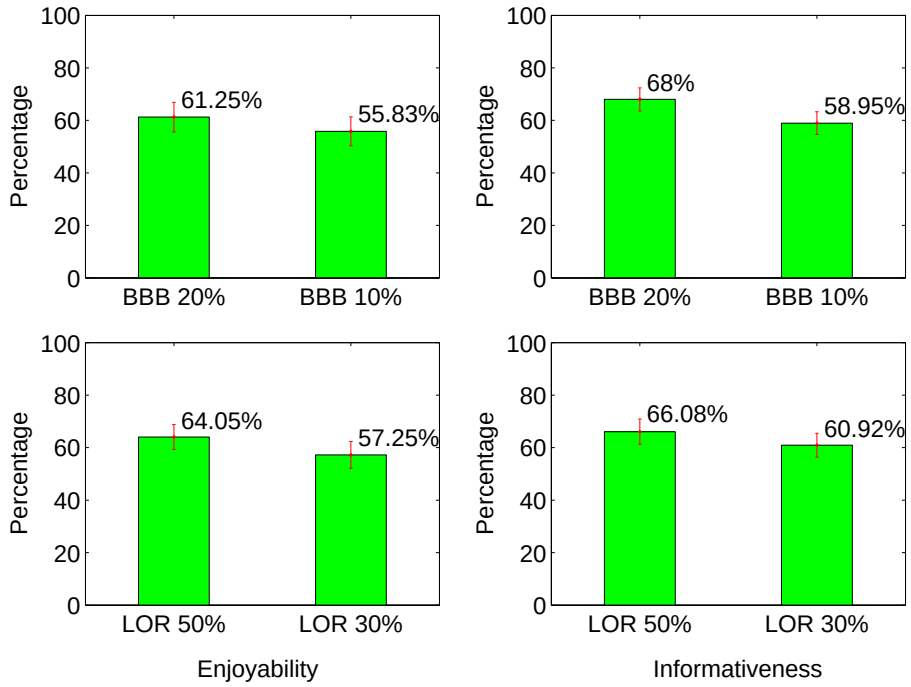


Figure 3-15. Statistical analysis results of the scores by subjective evaluation.

CHAPTER 4
A GENERIC FRAMEWORK FOR CONTENT-AWARE VIDEO CODING

4.1 Introduction

With the rapid development of video and multimedia technologies, digital video application has become one of the hottest topics which affect people's lives. The demand for digital video communication, such as video-conferencing, mobile broadcasting and videophone, has increased considerably thanks to the success of advanced video coding techniques, such as H.264 [54], MPEG-4, etc. However, due to the scarce of channel resource and the restriction of transmission rates, encoding video sequences at very low bit-rate with good quality remains a major challenge.

4.1.1 Content-Unaware Video Coding

At the same time, most state-of-the-art video coding standards, including H.264/AVC, treat each of their coding units (i.e., Macroblocks) equally. Although different macroblocks within the same frame may be coded with different modes and be partitioned into different sub-blocks, no one macroblock is more important than another, so no one will be favored for resource allocation. This model is easy and efficient, but is not always desirable when resources are really limited.

4.1.2 Existing Works of Content-Aware Video Coding

Extensive phycological studies reveal that, the human perception on an image (or, a video frame) is not "flat", i.e., some regions on the image may incur higher human attention compared to other regions. This natural phenomenon motivates people to design a 'smart' strategy for resource allocation. That is, when resources are limited, it is wise to sacrifice the bits consumption on inattentive regions and save these bits for salient regions. For example, when two people are making a video phonecall in two different places, under a pool wireless network, they may want to see each other's face more clearly, not the background. Thus, to encode the human faces with higher quality and the background with lower quality will satisfy the user's

need in such a circumstance. The regions that cause people's interests or convey more information, are named Region-of-Interest (ROI). For example, the speaker's face in a video-conferencing is an ROI other than the background; an anchorman or an anchorwoman who is broadcasting news is the ROI other than the studio; two table-tennis players in a game are ROI's other than the playground and spectators; a running boat in the river is the ROI other than the water and riverbank. Meanwhile, as human's eyes are more sensitive to motions, those regions with severe motion are very likely considered ROIs. There are two aspects that ROI can help to improve the performance of the existing coding standards. The first aspect is to improve error resilience capability. The second aspect is to improve the coding efficiency. This means to achieve better quality under the same bitrate budget or to achieve lower bitrate under same quality constrain, which is represented as:

$$\begin{aligned} \max \quad & PSNR \textbf{ s.t. } Rate \leq R_{target} \\ \min \quad & Rate \textbf{ s.t. } PSNR \geq PSNR_{target} \end{aligned} \tag{4–1}$$

In this chapter we mainly address the second aspect. Researchers have proposed many algorithms to do bit-allocation. Karlsson et al. use spatial-temporal filters [55] which forces undesirable background skip (reduce the background frame rate).Lin et al. use a frame-skipping scheme to do resource allocation in video conferences [56]. Chen et al. solve this problem as an optimizaiton problem by Lagrange theory [57]. Wang et al. design an algorithm which update the R-Q model to adaptively find the best quantization parameter [58]. Douglas Chai and King N. Ngan et al. [59] proposed two strategies, namely, Maximum Bit Transfer (MBT) and Joint Bit Assignment (JBA). In MBT, the largest QP is assigned to non-ROI and ROIs are optimized by the remaining bits. However, a poor background is not always desirable. JBA overcomes this drawback but still cannot avoid the abrupt quality degradation between ROI and non-ROIs.

4.2 Group-of-Picture Based Bit Allocation Framework

Consider the problems of existing work, we introduce a new framework for content-aware resource allocation. The coding mode of a macroblock in H.264 can be chosen from the set Mode = $\{INTRA4 \times 4$, $INTRA16 \times 16$, $INTER16 \times 16$, $INTER16 \times 8$, $INTER8 \times 16$, $INTER8 \times 8$, $INTER8 \times 4$, $INTER4 \times 8$, $INTER4 \times 4$, SKIP, DIRECT$\}$. For each macroblock S, the mode is first done to find the optimal RD_{cost} by minimizing

$$RD_{cost}(S, Mode|QP, \lambda) = D_{REC}(S, Mode|QP) + \lambda R_{REC}(S, Mode|QP) \tag{4–2}$$

where D_{REC} is the distortion, generally represented by the sum of the squared differences (SSD) or the sum of absolute differences (SAD), R_{REC} is the bits consumption for entropy coding on encoding the residual of the macroblock. λ is the Lagrangian multiplier, and QP is the quantization parameter.

$$\lambda = 0.85 \times 2^{(QP-12)/3} \tag{4–3}$$

The distortion D and rate R are also functions of QP. Many researchers have proposed many models to describe their relationship. For example, TMN8 suggests a quadratic model for the rate and distortion with QP, and some other people proposed their linear models instead [58]. Those prior works implies the fact that the adjustment of the quantization parameter QP is an effective method to do bit-allocation.

Considering a two-level bit-allocation scheme based on GOP (Group-of-pictures) structure as Fig. 4-1 shows. Since the first intra frame is the reference frame for motion estimation of succeeding inter frames, its quality is the dominant factor for the PSNR of the GOP. Thus, the compensation of bit-consumption between I-frame and P-frame will bring us benefits on PSNR improvement under same bit target. At the same time, the unimportant regions in P-frames (non-ROIs) are sacrificed to an acceptable extent.

Fig. 4-2 shows the encoder diagram for our ROI-based bit-allocation scheme. Compared to conventional H.264 coding diagram, we add a new module “Encoder Control”. In this module, the Group-of-picture structure are first formed. Then, through computation on contents of the Intra and Inter frames in the GOP respectively, an ROI flag will be assigned to each macroblock. When encoding the macroblock, the quantization parameter will be adjusted accordingly for quality control.

4.3 Intra-Frame ROI Identification and Bit-Allocation

Intra-frame ROI identification can be regarded as image-ROI identification, which can be solved by many techniques, such as skin color detection [60], level set segmentation [61], feature-based saliency detection [62] and transform-based saliency detection [2] etc. The discussion of image ROI-detection is can be found in chapter 2, where the saliency analysis is presented.

In our framework, we assume the ROI for intra-frame is already identified. Correspondingly, an ROI mask at macroblock level is generated. For simplicity, we assume all ROI-macroblocks are equally weighed. We should noticed the weight can also be measured by the relative importance. The bit-allocation scheme tried to take the advantage of ROI information and allocate more resource to ROI-macroblocks. This is realized by adjusting the QP for macroblocks. As we know, smaller QP reflects finer quantization and will cause smaller quantization error but at the same time, the bit consumption increases. Some notations: the actual QP for current macroblock is denoted $MB->currentQP$, the initial QP configured by user is QP_{ini}. The algorithm is described in Algorithm 7.

4.4 Inter-Frame ROI Identification and Bit-Allocation

Inter-frame ROI, however, is defined different to Intra-frames ROIs. This is due to the fact that Inter-frames utilize the reference frames to achieve coding efficiency. As we know, if the contents in a macroblock do not change much between an Inter-frame and its reference frame, the advanced video coding algorithm tends to use a SKIP mode

```
input  : Intra frame f_k to be encoded,Initial QP QP_ini, number of Macroblocks
         N_MB.
output: Encoded frame f_k, Bit ConsumptionB_k for f_k.
Do saliency detection on f_k;
for i ← 1 to N_MB do
    compute ROI flag R_i for Macroblock M_i;
    QP_{M_i} ← QP_ini;
    if R_i == 1 then
        M_i is an ROI;
        QP_{M_i} ← QP_{M_i} − −;
    end
    encode M_i with QP_{M_i};
end
finish encoding f_k;
compute B_k;
```

Algorithm 7: Proposed algorithm for Intra-frame ROI identification and bit-allocation.

for that macroblock, especially in low rate cases. This means the macroblock has no residue data so does not need to be quantized. For reconstruction, it only needs to copy the data from the reference frame with the corresponding motion vectors. Under such condition, the adjustment of quantization parameter (QP) is improper and meaningless. On the other hand, if the contents in a macroblock change a lot, that macroblock cannot be skipped and the residue data will be quantized. So we define that macroblocks with high motion activities are ROIs. The motion activities are measured by the statistics of motion vectors in a macroblock. The encoding QP control algorithm for Inter-frame can be described in Algorithm 8, where n is the pixel index in a macroblock, N is the total pixel number in a macroblock. The threshold T is updated such that the final bitrate for a GOP is within 3% fluctuation of the JM coded bitrate.

We should be aware of the fact of “quality propagation”, which is the phenomenon that, if the reference frame of an Inter-frame is in good quality (i.e., high PSNR), it is likely that the corresponding Inter-frame quality will also be good, especially when SKIP mode is used. This is due to the copying of macroblock data from the reference frame.

```
input  : GOP bits resource C, Bit Consumption of the Intra frame in the GOP B,
         Motion activity threshold T, Inter frame f_k to be encoded,Initial QP QP_ini,
         number of Macroblocks N_MB
output: Encoded frame f_k, Bit ConsumptionB_k for f_k
Do motion estimation on f_k;
for k ← 1 to N_{P-frames} do
    for i ← 1 to N_MB do
        compute motion activity MA_i for Macroblock M_i;
        MA_i = (1/N) Σ_{n=1}^{N} (|MV^x_{i,n}| + |MV^y_{i,n}|);
        QP_{M_i} ← QP_ini;
        if MA_i < T then
            M_i is not motion active;
        end
        QP_{M_i} ← QP_{M_i} + +;
        encode M_i with QP_{M_i};
    end
    finish encoding f_k;
    compute B_k;
    update T such that Σ_k B_k + B = C;
end
```

Algorithm 8: Proposed algorithm for Inter-frame ROI identification and bit-allocation

The "quality propagation" property gives us a hint to improve the P-frame quality while taking no extra overheads.

4.5 Experimental Results

The performance of region-of-interest-based bit allocation scheme is evaluated in this section. Standard test sequences with QCIF (176×144) resolution are carried out in our experiments. The simulation is implemented with the latest JVT reference H.264/AVC software JM14.0.

The experiments are designed as follows: benchmark sequence "Carphone.qcif" and "Akiyo.qcif" are selected. Both of them have a distinguished ROI (the human face), but "Carphone" has a varying background while "Akiyo" has a still background. Three bitrate levels are considered at frame rate 30fps for each test sequence: low (bitrate $<$ 32kbps), medium (32kbps $<$ bitrate $<$ 256kbps), high (bitrate $>$ 256kbps). For each sequence, 90 frames are tested. The 90 frames are divided into three group-of-pictures

(GOP) with Intra period 30 frames. Table 4-1 lists the experimental results. It is shown that under the same bitrate level (fluctuation less than 3%), the ROI-based coding scheme achieves better PSNR performance compared to JM14.0 baseline profile with the same GOP structure. The PSNR gain increases with the decrease of bitrate.

Table 4-1. Performance evaluation on benchmark sequence “Carphone” and “Akiyo”.

Test sequence	Bitrate (kbps)	PSNR(dB)		Avg. PSNR Gain (dB)
		JM	ROI	
Carphone	530	45.05	45.07	+0.02
	155	38.52	38.55	+0.03
	26	30.09	30.29	+0.20
Akiyo	433	50.98	51.00	+0.02
	117	44.21	44.33	+0.12
	12	30.85	31.10	+0.25

Fig. 4-3 shows the PSNR performance by frame for car.qcif under the low bitrate case. Fig. 4-4 depicts the PSNR gain of ROI-based coding scheme over standard JM. It is seen that the maximum PSNR gain is larger than 0.5dB and the average gain is 0.2dB. Fig. 4-5 shows the PSNR performance by frame for akiyo.qcif under the low bitrate case. Fig. 4-6 depicts the PSNR gain of ROI-based coding scheme over standard JM. Again, it is seen that the maximum PSNR gain is larger than 0.6dB and the average gain is 0.25dB. While achieving a higher PSNR, our ROI-based coding scheme simultaneously achieves a better visual quality. This can be seen from Fig. 4-7. To show the improvement of subjective quality for arbitrary frames, we pick up the first frame (Intra-frame and the first frame in a GOP), the 45th frame (P-frame and the middle frame in a GOP) and the 90th frame (P-frame and the last frame in a GOP). The first row shows original frames. The second row shows reconstructed frames by JM. The third row presents reconstructed frames by ROI scheme. The fourth row is the detail comparison of JM and ROI. Obviously our scheme has a better quality at the human’s eyes and mouse.

4.6 Summary

We have proposed a content-aware video coding framework that considers human vision properties. The framework utilizes the intrinsic relationship between intra and inter frames, resources are smartly allocated to salient regions, resulting in both subjective and objective quality gain over JM14. Our scheme also overcomes the drawback of abrupt ROI/non-ROI quality degradation of existing works. Meanwhile, it is easy to implement and totally compatible with standard H.264 decoders. The simplicity and good performance make it a promising solution for low-rate real-time video applications. The demo of this work can be found at: http://plaza.ufl.edu/lvtaoran/ROI.htm

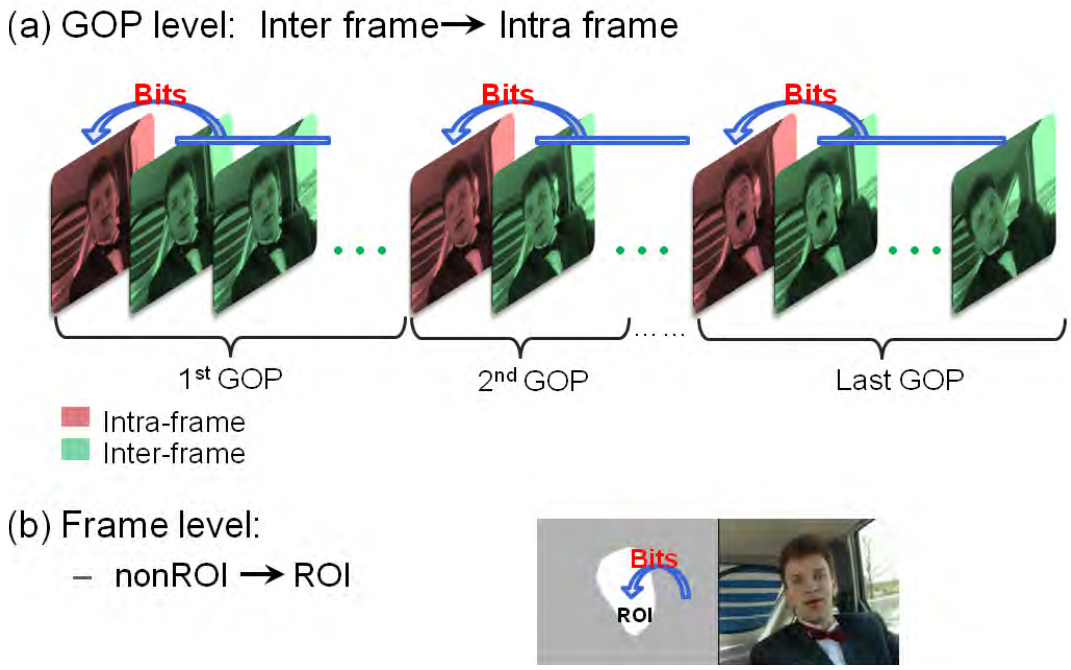


Figure 4-1. Two-level bit allocation. (a) GOP Level. (b) Frame Level.

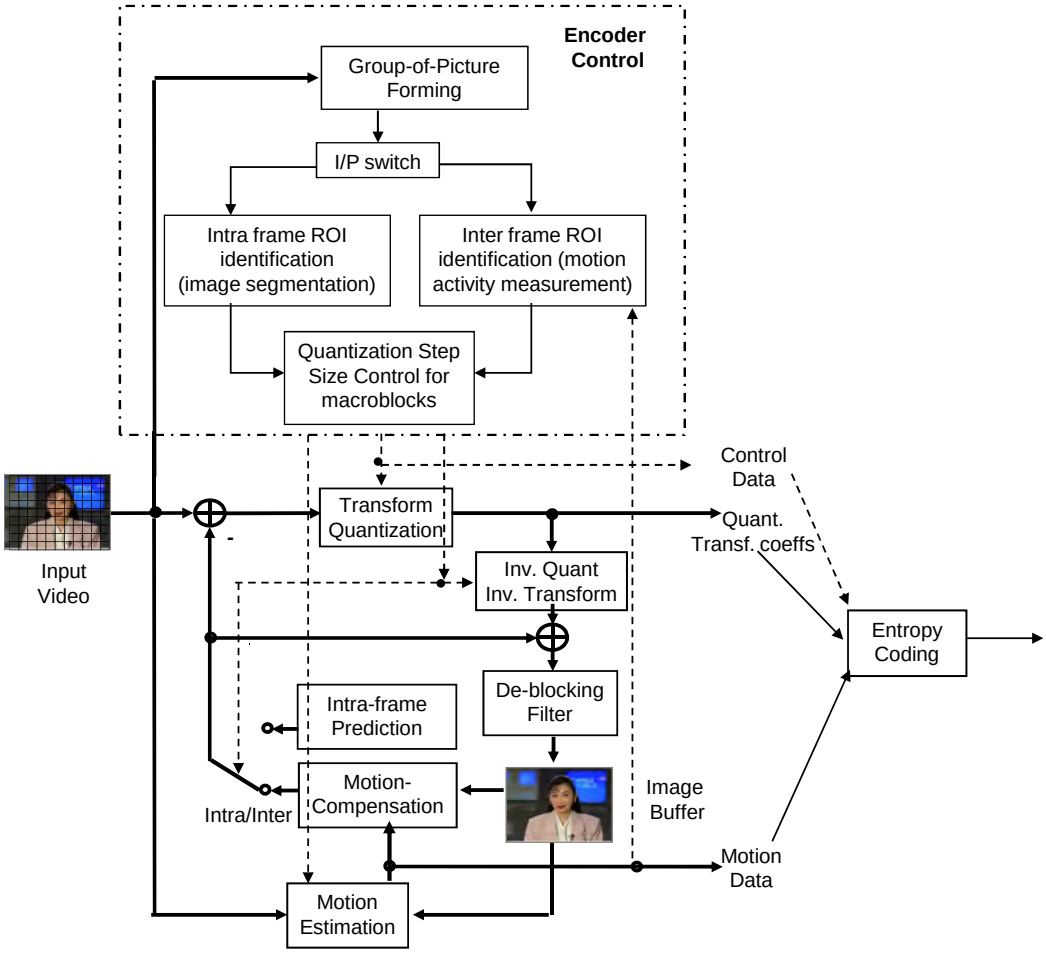


Figure 4-2. ROI-based encoder diagram.

Figure 4-3. PSNR comparison under low rate case for “Carphone”.

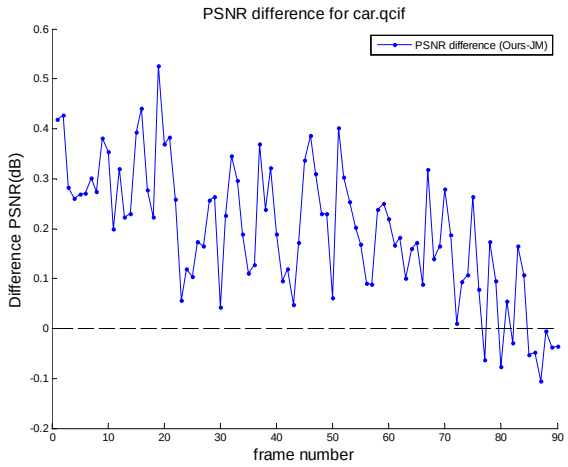


Figure 4-4. PSNR gain for “Carphone”.

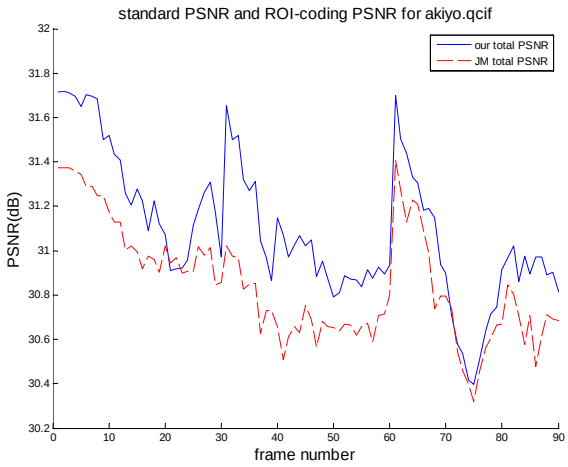


Figure 4-5. (PSNR comparison under low rate case for “Akiyo”.

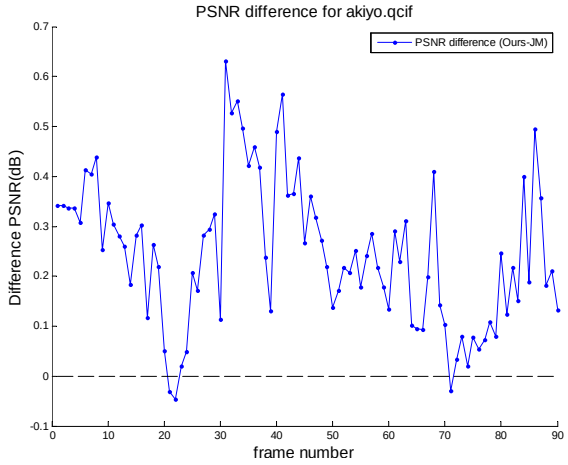


Figure 4-6. PSNR gain for “Akiyo”.

Figure 4-7. Reconstructed video frames of “Carphone” and “Akiyo” by ROI-scheme compared to JM14. (a) Carphone: Original video frame #0, #44, #89. (b) Carphone: Reconstructed frame #0, #44 and #89 by JM14. (c) Carphone: Reconstructed frame #0, #44 and #89 by ROI-scheme. (d) Detail comparison of (b) and (c). (e) Akiyo: Original video frame #0, #44, #89. (f) Akiyo: Reconstructed frame #0, #44 and #89 by JM14. (g) Akiyo: Reconstructed frame #0, #44 and #89 by ROI-scheme. (h) Detail comparison of (f) and (g).

CHAPTER 5
TRANSITION-BASED INTRA CODING

5.1 Introduction

How to effectively utilize spatial correlation is fundamental to the efficiency of current video codecs for intra coding. The state-of-art compression standard H.264/AVC [63] is the first video coding standard that employs spatial directional prediction for intra coding. It provides a flexible prediction framework, thus the coding efficiency is greatly improved over previous standards where intra prediction was done only in the transform domain. In H.264/AVC, spatial intra prediction is performed using the surrounding available samples, which are the previously reconstructed samples available at the decoder within the same slice (Fig. 5-1). The encoder typically selects the prediction mode that minimizes the difference between the prediction and original block to be coded. RD costs are calculated for several pre-defined directions and the best prediction mode is thus selected as the one with the least RD cost. The selected mode is then coded and transmitted to the decoder.

5.1.1 Existing Works

Although the intra prediction in H.264 can exploit some spatial redundancy within a picture, the prediction only relies on pixels above or to the left of the block which have already been encoded. The spatial distance between the pixels serving as predictions (which we call predictor pixels) and the pixels being predicted (which we call predicted pixels), especially the ones on the bottom right of the current block, can be large. With a large spatial distance, the correlation between pixels can be low, and the residue signals can be large after prediction, which affects the coding efficiency. In addition, extrapolation is used instead of interpolation because of the limitation of causality.

In [64], a new encoding method for the planar mode of intra 16×16 is proposed. When a macroblock is coded in planar mode, its bottom-right sample is signaled in the bitstream, the rightmost and bottom samples of the macroblock are linearly interpolated,

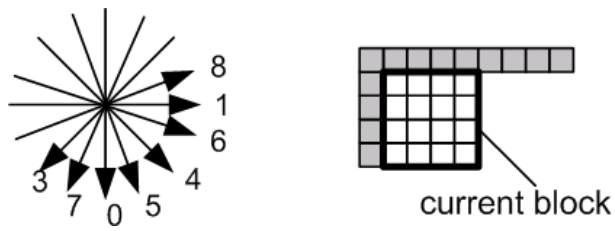


Figure 5-1. H.264 Intra prediction modes for 8×8 and 4×4 blocks.

and the middle samples are bilinearly interpolated from the border samples. When planar mode is signaled, the same algorithm is applied to luminance and both chrominance components separately with individual signaling of the bottom-right samples (16×16 based operation for luminance and 8×8 based for chrominance). The planar mode does not code the residue. Although the new planar prediction method exploits some spatial correlation with the bottom-right sample, the prediction accuracy of the right and bottom pixels are still quite limited.

In [65–67] of the key technique area (KTA) [68], bidirectional intra prediction (BIP) is proposed to improve the intra coding efficiency. Two features are proposed: one is the bidirectional prediction that combines two unidirectional intra prediction modes, and the other is the change of the sub-block coding order in a macroblock. By introducing the bidirectional prediction, BIP increase the total number of prediction modes from 9 to 16. To change the sub-block coding order, it encodes the bottom-right 8×8 (or 4×4) sub-block first before encoding the other three sub-blocks. Whether to change the coding order is an RD cost based decision which needs to be signaled to the decoder. Although the BIP method greatly improves the coding efficiency, the complexity of this algorithm is very high: H.264 loops over 9 modes for 8x8 blocks while BIP has to loop over $16 \times 2 = 32$ modes to select one with the minimum RD cost. BIP also requires more bits to signal the mode and coding order.

5.1.2 Overview of Our Approach

In this chapter, we propose a new method to encode an intra block. The proposed method derives the intra prediction direction of a block or a block partition using its

surrounding pixels. Therefore, no mode selection is needed so syntax bits are saved, and prediction directions other than the 9 pre-defined directions are allowed.

We name our algorithm TIP for abbreviation of transition-based intra prediction. “Transition” here stands for the altering between black and white pixels on a binary mask. Zeng [69] proposed a geometric-structure-based directional filtering scheme for error concealment of a missing block, where the boundary information is always available. We simplify Zeng’s analysis for transition cases, extend it for intra prediction and incorporate it with Shiodera’s new coding order framework in [65–67] to make the block boundary available. Experiments show that TIP can achieve up to 10% bitrate savings over JM.

To facilitate the development of new coding tools, we also design a novel video parameter analyzer as the side product of this work. This analyzer is independent of bitstream syntax so it can be widely used. We include the introduction of the analyzer in this chapter too. The rest of this chapter is organized as follows. In Sec. 5.2, we first introduce the method to generate the transition points on block boundaries. Then we present the algorithm to analyze local geometric patterns, and show the interpolation scheme in TIP mode. Then we discuss the new coding order scheme to support this new mode in Sec. 5.3. Experimental results of the transition based intra coding are presented in Sec. 5.4. The design principle and implementation details of the analyzer are presented in Sec. 5.5. Finally, we conclude our work in Sec. 5.6.

5.2 Transition Cases and Interpolation Schemes

The sudden change of neighboring pixel values forms a transition. Huang and Algazi have shown in their work [70, 71] that within a small analysis window of an image/frame, the local geometric structure can often be characterized by a bimodal distribution. Thus, a transition from black to white (or vice versa) reveals the existence of an edge. Given the transition distribution on a block boundary, we can analyze the

local geometric patterns within the block. Intra prediction is thus benefit from the local geometric patterns.

As we know, a line is defined by two points. In order to find the local geometric structure along a block boundary, two nearest surrounding boundary layers are examined. The transition points on inner layer indicate the location of an edge and the transition points on the outer layer help to determine the angle of that edge. The two layers are first converted into a binary pattern, and the threshold for binarization is adaptively calculated. Several methods can be used for calculating the threshold, including the simplest mean pixel value of boundary layers, the average of the fourth largest value and the fourth smallest value used in [69], and most complicated histogram based segmentation. After binarization, a three point median filter is applied to eliminate isolated black or white points.

We define the white point corresponding to a transition (black to white, or white to black) as a *transition point*. As shown in Fig. 5-2, the red dots on inner layer and dark-red dots on outer layer indicate the transition points. Note that since the boundary is a closed loop, the number of transition points is always even.

A transition point on the inner layer implies a local edge, while its corresponding transition point on the outer layer helps to identify the slope of the edge. Depending on the number of transition points on the inner layer, the situation is classified into four cases: flat (0 transition), 2, 4 and more than 4. A measure of directional consistency is used to resolve the ambiguity about how the transition points on the inner layer should be matched to each other to illustrate the local edge structure.

In the clockwise direction, for the $i-th$ transition point on the inner layer, denote the angle of the line connecting this point and its corresponding transition point on the outer layer as θ_i (see Fig. 5-2). The angle of the line connecting the $i-th$ transition point and the $j-th$ point on the inner layer is denoted as θ_{ij}. An assumption for the local geometric pattern is: If there is an edge passing through transition points i and j, then

θ_{ij}, θ_i and θ_j should be consistent. The measurement for consistency is defined as

$$C_{ij} = |\theta_i - \theta_{ij}| + |\theta_j - \theta_{ij}| \tag{5–1}$$

5.2.1 Flat/zero Transition Case

We will discuss the decision rules and interpolation schemes for each case in this section. When the binarization threshold is too close to the maximum and minimum value, or the local variance is relatively small, the current block is a smooth block. In this case, the projective interpolation scheme of [72] is used. In this scheme, the edge orientation is classified into eight possible directions, i.e., $k \times 22.5°, k = 0, 1, ..., 7$. Let P_k^1 and P_k^2 represent the two sets of projection data at orientation k. The best orientation is found with minimizing the projection difference:

$$k = \arg\min_{0 \le k \le 7} diff(P_k) = \arg\min_{0 \le k \le 7} \frac{|P_k^1 - P_k^2|^2}{Dim(P_k)} \tag{5–2}$$

Given the orientation, the intra predictors $I(p)$ at pixel p can be generated by bilinear interpolation along that orientation k:

$$I_p = \frac{d_2}{d_1 + d_2} p_1 + \frac{d_1}{d_1 + d_2} p_2 \tag{5–3}$$

where p_1 and p_2 are linearly interpolated from their two nearest neighboring pixels on the inner layer, and d_1, d_2 are the Euclidean distances of p with p_1 and p_2.

5.2.2 Two Transitions Case

For two transition points, there are two conditions. The first condition is that an edge goes through the two transition points (Fig. 5-3). This is the most likely case. The other is that a streak or corner exists (Fig. 5-4). The interpolation schemes are slightly different for these two conditions. The decision is based on Eq. 5–1: If $C_{01} < 3\pi/4$, then an edge exists. The predictors are generated using bilinear interpolation along θ_{01}. Otherwise, a streak or corner exists and the interpolation is along $\theta = (\theta_0 + \theta_1)/2$.

5.2.3 Four Transitions Case

Four transitions case is more complex than two transitions case. Denote the transition points starting from top in the clockwise direction as 0, 1, 2, 3 on the inner layer. There are several situations. 1) $C_{01} + C_{23} < C_{03} + C_{21}$, and $C_{ij} \neq \pi$. When this is true, it is assumed that transition point 0 is connected to transition point 1. 2) $C_{01} + C_{23} \geq C_{03} + C_{21}$, and $C_{ij} \neq \pi$. In this situation, transition point 0 is connected to transition point 3 (Fig. 5-5). For these two situations, the two edges divided the block into three regions. The bilinear interpolation for each pixel is along the direction of the edge that is closer to the pixel. 3) Sometimes C_{ij} is close to π. It is assumed a strong edge with another narrow streak goes into and stops in the block (Fig. 5-6). In this case, every pixel is first bilinearly interpolated along the direction of the edge, then the pixels in the streak are interpolated along the direction of the streak.

5.2.4 Six or More Transitions Case

When six or more than six transition points are found, to discover the optimal combination of edges is complex and difficult. In practical, we analyze the distribution of transition cases with several bench mark video sequences. The result shows that cases with six and more transition points are rare. An example is shown in Fig. 5-7 that only less than 5% of the blocks have six and more transition points. Thus, a simple interpolation scheme is used for this case without severely degrade the overall performance: we select the most frequent direction θ_f among all θ_i as the dominant direction. All the predictors are generated bilinearly along that direction.

With the aforementioned transition point analysis and interpolation schemes, we are able to generate all the predictors for intra prediction. The residues are then encoded and sent to the bitstream. Since the prediction direction is derived by the algorithm itself, no syntax bit is need to explicitly signal the TIP mode.

5.3 New Encoding Order

We have discussed the algorithm to generate the intra predictors in the previous section, which is based on the assumption that all the surrounding pixels of a block are available. However, in current H.264/AVC coding framework, this is not the case. Only the blocks at the top or to the left of the current block are available with the raster encoding order. In order to make all surrounding pixels available for some blocks, we incorporate the reverse coding order from [67].

We take the encoding process of four 8×8 blocks in a macroblock as an example to illustrate how the reverse coding order works with TIP. As shown in Fig. 5-8, the left figure shows the raster coding order, and the right figure shows the reverse coding order. The bottom right (BR) block will be encoded first using the top and left neighboring macroblock pixels (the region in grey). Next, the upper right (UR) block is encoded using the top and left neighboring macroblock pixels and the reconstructed BL block as well. Then, the bottom left (BL) block is encoded using the top and left neighboring macroblock pixels, the BR and UR block. Finally, the upper left (UL) block is coded by TIP mode with all its surrounding pixels available. The prediction modes for each block are shown in Fig. 5-9. The encoder will choose the coding order with corresponding modes under the rate-distortion optimization criteria.

$$J(Mode) = D + \lambda * R \tag{5–4}$$

$$Mode = \arg\min J(i) \tag{5–5}$$

where D is the MAD of source and predicted pixels, which is the measure of distortion under a mode. R is the bit rate incurred to code the mode syntax and the residue under the mode. λ is the Lagrange multiplier.

5.4 Experimental Results

We implement the intra prediction algorithm on KTA 1.4 and test its performance using the MPEG new call-for-proposal sequences, which contain three sets of

Table 5-1. Experimental results of TIP over JM.

Sequences (Name_resolution_framerate)	Bitrate saving(%)	PSNR Gain (dB)
BasketballDrive_1920x1080_50	10.30	0.31
BQTerrace_1920x1080_60	5.46	0.37
cactus_1920x1080_50	4.21	0.18
Kimono1_1920x1080_24	6.65	0.25
ParkScene_1920x1080_24	4.04	0.18
BasketballDrill_832x480_50	4.31	0.21
BQMall_832x480_60	5.47	0.35
PartyScene_832x480_50	3.49	0.28
RaceHorses_832x480_30	2.51	0.17
BasketballPass_416x240_50	5.93	0.35
BlowingBubbles_416x240_50	3.19	0.20
BQSquare_416x240_60	1.47	0.13
RaceHorses_416x240_30	2.82	0.20
Average	4.60	0.24

sequences with resolution 1920×1080, 832×480, and 416×240. The configuration parameters of the experiments are: Frame to be encoded: 50. Coding structure: IIIII. Test points: $QP = 22, 27, 32, 37$. The results are listed in Table 5-1. The rate-distortion curve of sequence "BasketballDrive_1920x1080_50" is shown in Fig. 5-10. Results show that the proposed approach achieves an average 4.6% bit rate saving over JM.

5.5 An Coding Parameter Analyzer

5.5.1 Introduction

A video codec analyzer is a powerful tool in designing video coding algorithms. Instead of taking time to read the log files containing the intermediate coding parameters, such as mode and motion vectors of every macroblock, video codec developers can gain more insights by looking at the graphic display of these information with the analyzer.This enables the users to quickly and easily check the conformance and encoding algorithm performance. Moreover, it can be very helpful in assisting and guiding the encoding algorithm design.

A typical video codec analyzer takes the encoded bitstreams as input and decodes it before visualizing the encoding parameters through a graphical user interface (GUI).

Commercial video analyzers have been developed in the past, such as Sencore CMA1820 [73], Elecard Stream Analzyer [74], etc. These analyzers are designed for syntax analysis and presentation of the encoding parameters in a visual form. They support MPEG-2, H.264/AVC [63], and VC-1 video formats along with other audio formats.

While various analyzers offer different GUI, they are all capable of displaying the decoded video, high-level syntax, and encoding parameters such as QPs, motion vectors, partition modes, reference indexes and warning or error message if necessary. For syntax conformance checking, the syntax information is often visualized in a tree structure, which enables the developers to easily verify new encoder compliance before deployment. In addition, an analyzer allows users to quickly identify the potential problems in the encoder algorithms because well-tuned encoding parameters are often highly correlated with video content. It becomes easier to identify inconsistencies and confirms the effectiveness of the coding algorithm one is interested in by displaying the encoding parameters and the video content in the same window. For example, a salient area should choose a smaller QP than an inattentive area to improve visual quality. When we overlay the QP for each macroblock (MB) on top of the picture, we can clearly see whether the QP is adaptive to the texture of that MB.

It is common to use a video bitstream analyzer when developing a new product of encoder or decoder that conforms to a standard, such as MPEG-2, H.264/AVC, VC-1, etc. The analyzer may provide picture-level coding parameters, such as the entropy coding method, picture-level QP, etc. Based on the user's request, the analyzer may display on the monitor block-level coding parameters, such as QPs, modes and motion vectors. An analyzer may also check the conformance of the bitstream to a certain standard and provide detailed warning or error messages if the bitstream is not conformant.

When the aforementioned video bitstream analyzer is highly appreciated during the product development after a standard is finalized, there is a strong need for a video analyzer that can be used during the standard development stage and is more resilient to the syntax changes. Thus, we propose such a video analyzer that will not take the coded bitstream, but takes the encoder/decoder statistical data as its input. Since the analyzer does not take bitstream as an input, the analyzer is independent of the bitstream syntax and can be used before the standard is finalized. Because of its independence of the syntax, such an analyzer can be used by different video encoding formats too. This kind of analyzer will be especially useful for developing the future video coding standard H.265.

Fig. 5-12 illustrates a typical video bitstream analyzer for H.264 [73, 74]. It requires a H.264 standard-compatible bitstream as its only input. Conformance check is performed and warning or error message is provided if there exists any problem in the syntax. The bitstream is then partially or fully decoded to obtain coding parameters such as the motion vectors, mode types, QP values, and/or residue sample values. Upon the user's request from the GUI, the analyzer displays the corresponding data on the monitor. For example, a user can request through the GUI to display the motion vectors for a frame.

As shown in Fig. 5-12, these analyzers contain an embedded video decoder that decodes the bitstream before visualizing it. They require the input bitstream to be completely compliant with the decoder syntax specifications. This is appropriate for designing standard-compatible encoder products after the standard has been finalized, but it is not desirable at the standard development stage. During the standardization process, many proposals compete for adoption, so the coding tools and syntax definitions change frequently. An ideal analyzer should be robust and flexible enough to accommodate different solutions. Motivated by this, we propose a novel video coding analyzer that decouples from the embedded decoder and instead takes the coding

Table 5-2. Four categories of coding parameters.

Frame-level parameter	frame type, QP, resolution extended MB size, QALF size
Extended MB-level parameter	MB mode, MB-level QP, CBP, filter flag
MB partition	extended MB partition, filter block partition
Sub-block level parameter	motion vector, reference index, intra prediction mode

parameters as the input. Then, the analyzer only parses the coding parameters and is syntax-insensitive. Thus, compared to the existing stream analyzers, our proposed analyzer offers more flexibility and can accommodate the constantly changing syntax definitions at the development stage of a new video coding algorithm or standard.

5.5.2 The Proposed Analyzer

Fig. 5-13 illustrates the framework of the proposed analyzer. It takes the coding parameter files and the reconstructed YUV video sequence as inputs before displaying them on the GUI. Since the coding parameter files and the YUV can be generated at either the encoder or the decoder without any knowledge of the syntax definitions, the presence of a decoder becomes optional in our analyzer (dashed box in Fig. 5-13), while it is mandatory for the H.264 bistream analyzer (Fig. 5-12).

5.5.3 Coding Parameter Files

In the following, we will explain how to generate the coding parameter files at the encoder (or the decoder) before describing the functionalities of each module of the analyzer.The coding parameter files from the encoder/decoder are taken as inputs of our analyzer. Given the different proposals [75] for next-generation video coding standards, we use the KTA software as our base codec and consider others to be similar. The essential parameters include coding modes, motion vectors, reference indexes, block partitioning for each coding unit (e.g., a macroblock), etc. Other parameters are new and may provide higher coding efficiency, such as the extended MB size and the quad-tree adaptive loop filter (QALF) [76] for KTA. The exact parameter values may still differ

among competing solutions. For example, the largest extended block size is 64 in KTA, and it may be 128 in another platform. Our analyzer is flexible enough to accommodate such differences.

To design an analyzer that is easily extendable, we study the properties of all these parameters and classify them into four categories, which are summarized in Table 5-2. Parameters of the same category are output at the encoder and parsed at the analyzer following the same methodology, which is explained in the following subsections. When a new coding tool is introduced, we can classify the related parameters into the categories and output them in the coding parameter file in a similar way.

1) Frame-level parameter.

The frame-level information includes the frame type, QP, resolution, extended MB size, QALF block size, etc. Such information is global to the frame and has a large impact on the visual quality. These data are written into a text file.

2) Extended MB-level parameter.

It appears that the next-generation video coding standard will still use the block as the basic coding unit within a frame. In the case of KTA, the extended MB is the basic coding unit. There are a few coding parameters at the extended MB level. For example, the INTRA/INTER coding mode is distinctive for each MB (in B frames it supports sub-MB direct modes, thus the minimum unit is an 8x8 block). We store the coding mode index in the original scanning order. It can be easily visualized for every MB or sub-MB, either through different colors or displaying the indexes. Other parameters in this category include the MB-level QP, the coded block pattern (CBP), the block-adaptive filter flag, etc.

3) Partition pattern within extended MB.

In KTA, the partition pattern within each super block can be different and needs to be input to the coding analyzer. Since the number of possible partitions is very large and it varies with the super block size, it is not realistic to represent the partition for a

super block by an index. Instead, we use a hierarchical representation. Let “1” stand for the current block having four equal-sized sub-partitions and “0” for the current block not being further partitioned. Assuming we use 64x64 extended MBs in KTA, the partition in Fig. 5-14(a) is represented by a binary string “1 1 0 0 0 0 0 0 0”, and Fig. 5-14(b) by “0”. Another example is the QALF, which uses a quadtree-based partition over each MB (the size of which needs to be signaled at a frame level) and the partition pattern needs to be signaled in the hierarchical fashion similarly. In our work, the data in this category are saved in a text file. Each line in the text file represents partition parameters for an extended block.

4) Sub-block level parameter

After the partition pattern within an extended MB is signaled, the sub partition-level coding parameters (e.g., the motion vectors and reference indexes for each sub-block) have to be represented. Take the motion vectors in H.264 and KTA as an example, each sub block has two sets of motion vectors: the list 0 and list 1 motion vectors. The vectors are stored for each block and can be easily visualized. Other parameters that belong to this category include the INTRA prediction mode. In our work, the data in this category are saved block by block in a binary file.

5.5.4 Analyzer Modules

As shown in Fig. 5-13, our analyzer consists of two modules: a data parser and a GUI. In this section, we explain the functionality of each module.

5.5.4.1 Data parser

The data parser reads the coding parameter data files and the YUV sequence. As the coding parameter data falls into different categories as defined in Section 5.5.3, we have corresponding parsing functions to extract coding parameters from the formatted data files, which are then organized into a matrix or a bitmap that is ready to be displayed. For instance, the partition patterns for the extended MBs are organized into a logical bitmap of the same size as the video frame; the lines where partition

happens are indicated by logic value TRUE (otherwise, FALSE is indicated). For the coding mode parameter, different modes correspond to different intensity values in the bitmap.

Another functionality of the data parser is to compute the distribution/statistics of the coding parameter data. For instance, it can collect statistics such as the frequency of each mode being selected and the percentage of different MB partition patterns within a frame. Such knowledge is greatly appreciated during the algorithm design. Take the mode distribution as example, the user can design a more efficient codebook based on the mode distribution with the frequently selected mode using a short codeword to save bit rate.

5.5.4.2 Graphical user interface

The GUI displays graphical objects based on the extracted data from the data parser. It also generates statistic plots and offers a set of interactive tools for users. In our work, the GUI presents the following information:

- coding parameters in Table 5-2
- statistical distribution of coding parameters
- the MB grid
- the Y/U/V channels
- the frame preview

The “on” and “off” of each information display can be controlled independently, and the information can be overlayed to form a comprehensive display, which can be continuously played as a video. The users can access an arbitrary frame by entering the frame number, get the detail data by clicking on a particular MB, or take a snapshot of the analyzed results for record. All these iterative functionalities make the proposed analyzer a convenient and effective tool in assisting and guiding the design of video coding algorithms.

5.5.5 Examples

We have output the coding parameter files and the decoded YUV from the KTA software and performed analysis with the proposed analyzer. Please note that our analyzer can be easily extended to analyze other competing video coding softwares. In this section, we illustrate how to use our analyzer by a few screenshots.

5.5.5.1 Main graphical-user-interface

Fig. 5-15 shows a snapshot of the main GUI window, which is an example of the KTA mode/partiton overlaying on the YUV. The control bar at the top provides easy access to all functionalities. The preview window at the right not only allows easy navigation among frames but also gives a simple preview of the neighboring pictures. The text window at the bottom shows the frame-level information together with the mode distribution. When a block is selected, its pertinent motion information is detailed in the text window.

5.5.5.2 Parameter display

Fig. 5-16 shows an example of the KTA QALF flag. Note that the overlay transparency can be adjusted from opaque to transparent. Fig. 5-17 illustrates the mode statistical distribution plot generated by our analyzer. The distribution is helpful in designing the codewords. For example, if a mode is frequently selected, a shorter codeword should be assigned. Fig. 5-18 presents an example for QP variation among MBs, and Fig. 5-19 shows the block partitioning and coding modes overlay.

5.6 Summary

In this chapter, we have proposed a novel intra prediction mode which utilizes the binary transition points on block boundaries to explore the local geometric pattern. The TIP mode derives the intra prediction direction and performs interpolation without transmitting syntax bits for this mode. A reverse coding order is introduced to make surrounding pixels available for TIP. Experimental results show a promising bit rate saving and PSNR gain over JM. However, constrained by the availability requirement of

surrounding pixels, the TIP mode is only allowed on a subset of the blocks, which is its major limitation.

As a side product, we proposed a video coding analyzer that excludes an embedded decoder and does not rely on syntax definitions. As MPEG and VCEG are now actively developing the next-generation video coding standard, a comprehensive video analyzer that visualizes the coding parameters is highly attractive to the developers. The analyzer can display the video as well as the coding parameters, including motion vectors, modes, partitions, filter regions, QPs, etc. It can also calculate and display the statistics of the input parameters. By integrating the video content and the coding parameters into one window, the analyzer provides a comprehensive review of performance of the coding tools. Moreover, since it is insensitive to syntax elements, the analyzer can be easily extended for other video coding softwares. Therefore, this is a convenient and powerful tool for the development of the next-generation video coding standards. Demos for the proposed analyzer are provided at website: http://plaza.ufl.edu/lvtaoran/analyzer.htm.

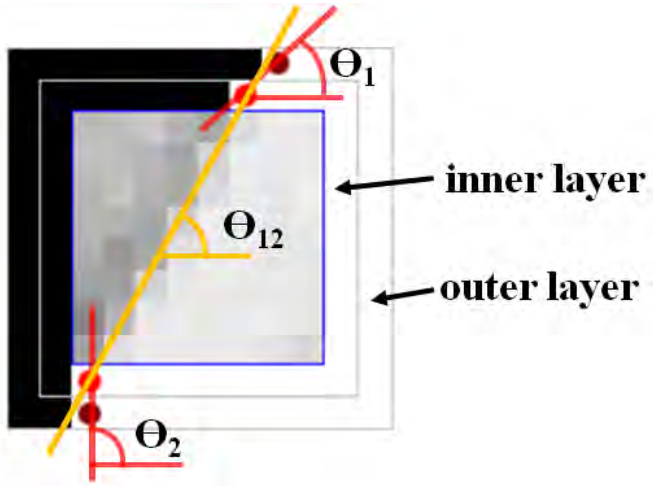


Figure 5-2. Transition points on inner layer and outer layer.

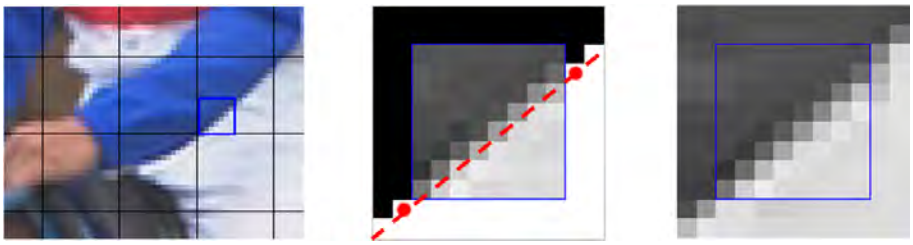


Figure 5-3. Two transitions case: An edge goes through. Left: Original. Middle: Transition points on boundary with original pixel values in block showing the local geometric patterns. Right: Original boundary with interpolated block pixels (within blue rectangle).

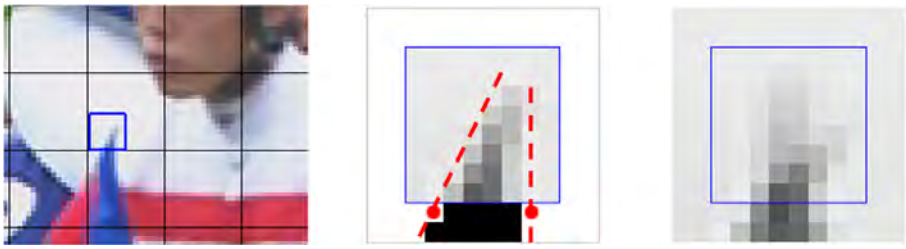


Figure 5-4. Two transitions case: A streak or corner.Left: Original. Middle: Transition points on boundary with original pixel values in block showing the local geometric patterns. Right: Original boundary with interpolated block pixels (within blue rectangle).

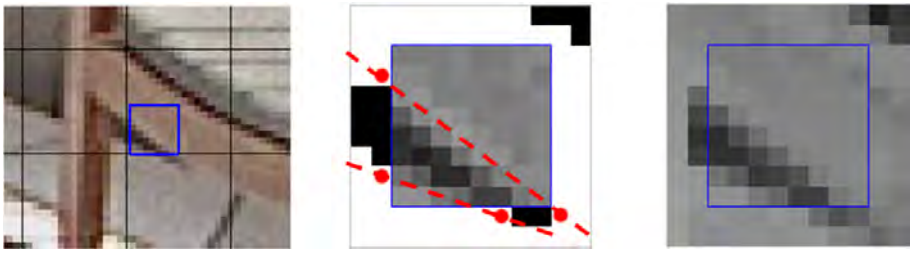


Figure 5-5. Four transitions case: transition point 0 is connected to point 3. Left: Original. Middle: Transition points on boundary with original pixel values in block showing the local geometric patterns. Right: Original boundary with interpolated block pixels (within blue rectangle).

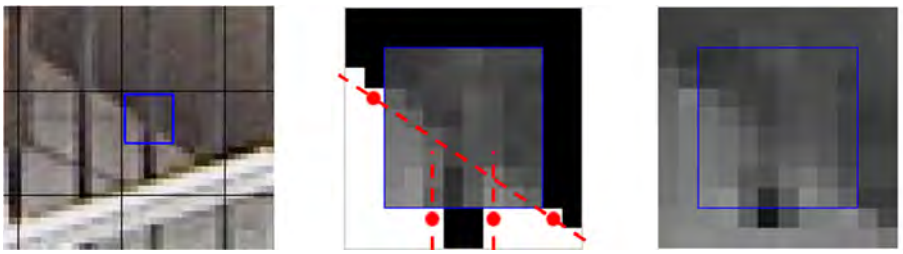


Figure 5-6. Four transitions case: An edge and a streak. Left: Original. Middle: Transition points on boundary with original pixel values in block showing the local geometric patterns. Right: Original boundary with interpolated block pixels (within blue rectangle).

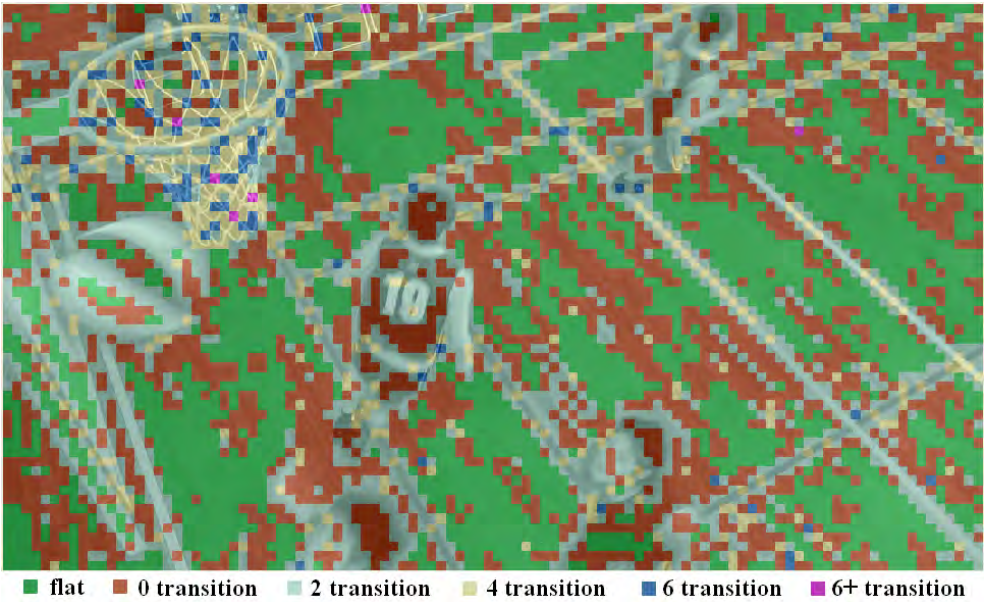


Figure 5-7. Distribution of transition cases of 8×8 blocks for BasketballDrill_832x480.

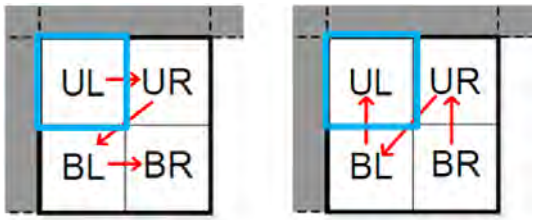


Figure 5-8. Raster coding order vs. reverse coding order. Left: Raster coding order. Right: Reverse coding order.

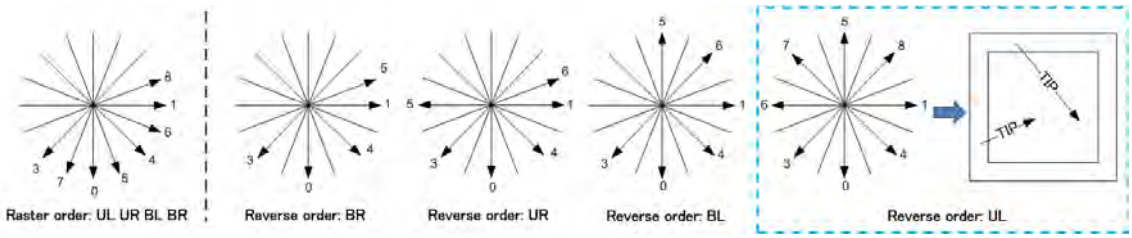


Figure 5-9. Intra prediction modes for BR, UR, BL and UL blocks with raster and reverse coding order.

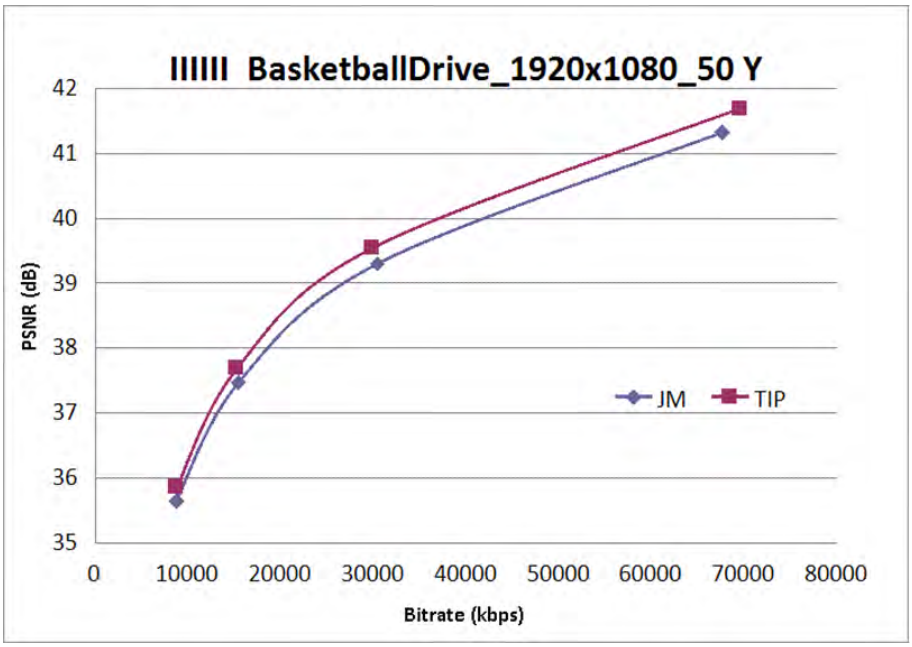


Figure 5-10. Rate-distortion curve for TIP vs. JM.

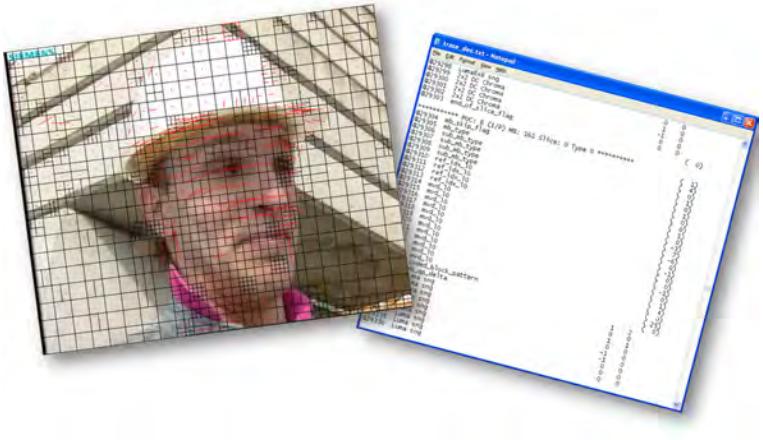


Figure 5-11. The motivation for using an analyzer instead of checking log files.

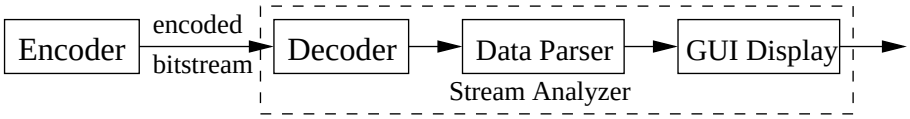


Figure 5-12. Workflow of a conventional H.264 bitstream analyzer.

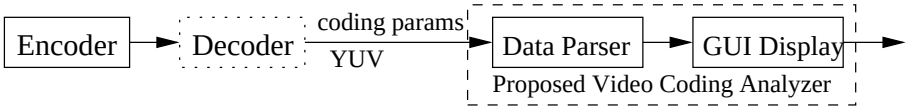


Figure 5-13. Workflow of the proposed video coding analyzer.

(a) (b)

Figure 5-14. Block partitions can be represented by binary strings: (a) “1 1 0 0 0 0 0 0 0” and (b) “0”.

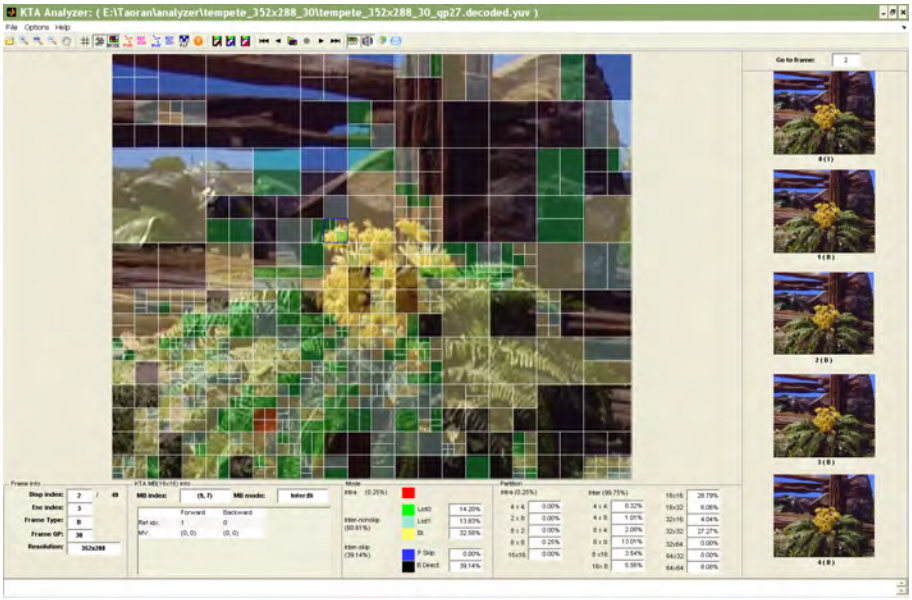


Figure 5-15. An example GUI screenshot.

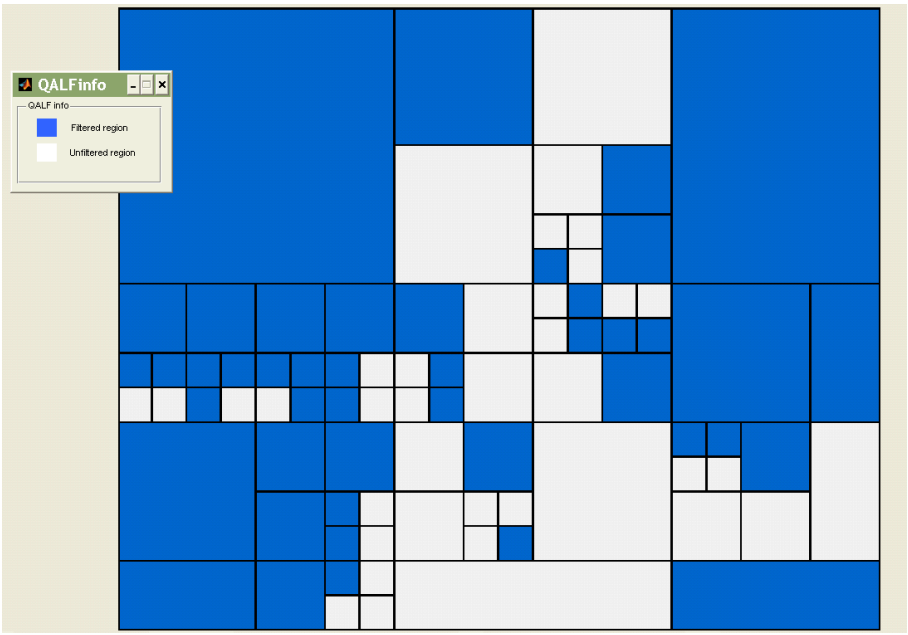


Figure 5-16. An example QALF filter information.

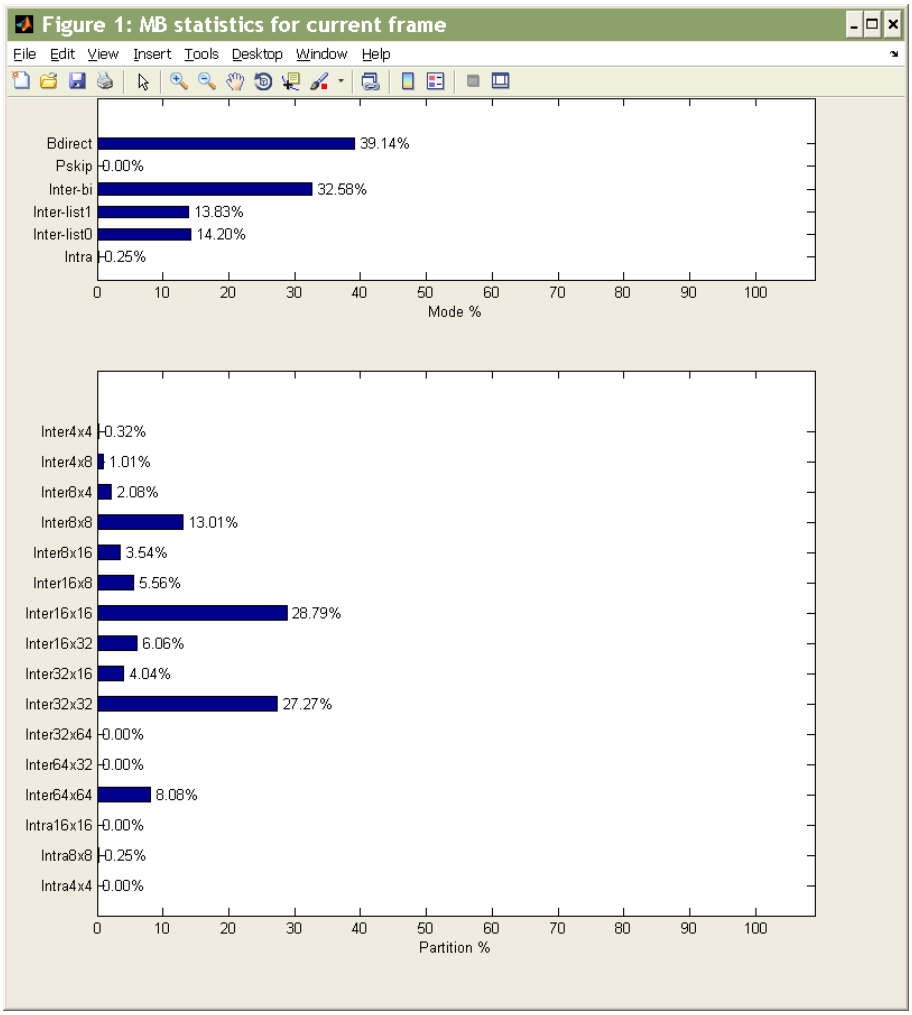


Figure 5-17. An example mode distribution plot.

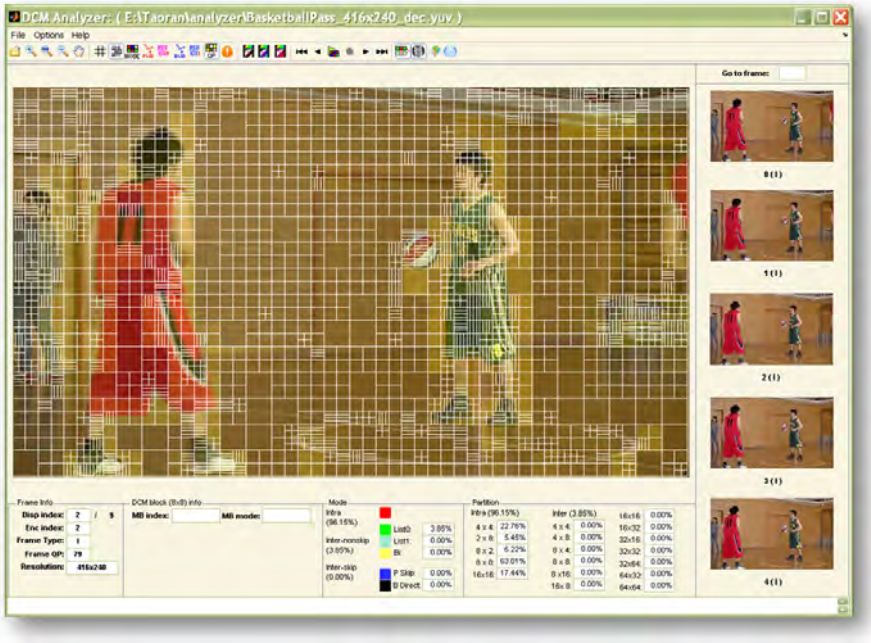


Figure 5-18. An example QP variation plot.

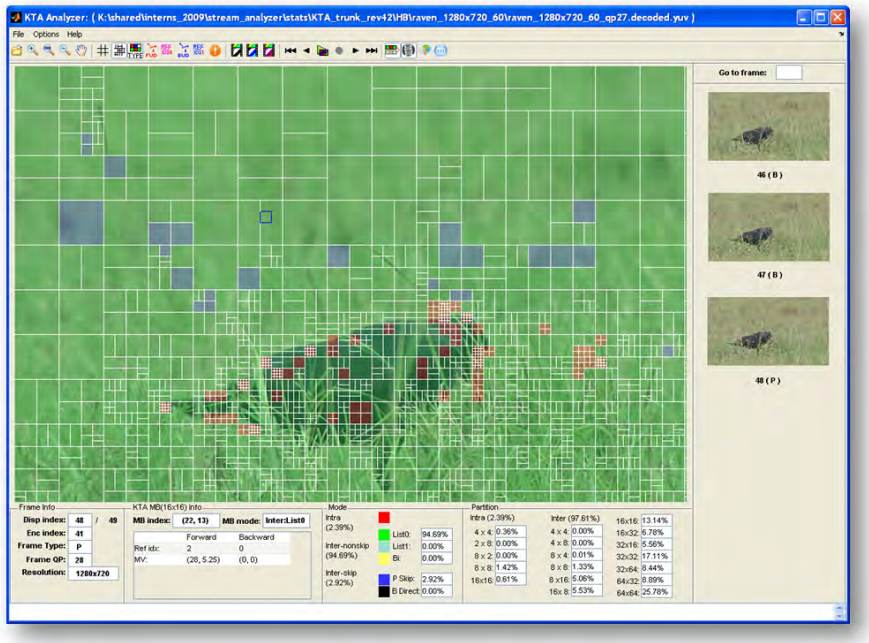


Figure 5-19. An example partitioning and mode overlay plot.

CHAPTER 6
CONCLUSION AND FUTURE WORK

In this dissertation, we have presented our work and innovations on three challenging problems of digital video applications: video compression, video summarization and video adaptation. We studied the modeling for video contents and proposed a nonlinear spatial-temporal saliency map for human attention modeling. With this attention model, we are able to establish content-aware techniques for solving these problems.

Video adaptation is conceptually a compression in the resolution domain. The goal is to fit an existing video with high resolution to an arbitrary display with low resolution. With the saliency maps, we proposed content-aware information loss metrics and formulated the content mapping problem as a shortest-path problem which is then solved by dynamic programming. The solution corresponds to the optimal parameters (x, y, s) for a cropping-and-scaling window over the whole video sequence, where (x, y) stands for the location of the window and s is the scaling factor. The cost function of the shortest-path problem has two terms, which stand for intra frame considerations and inter frame considerations, respectively. The weighting factor ω to balance the two terms is currently adjustable upon user's preference. To find the optimal choice of ω is one of our future directions. Another future work is to refine the scientific evaluation method to quantify and compare the performance of the algorithms which generate subjective results.

Video summarization can be regarded as a compression in the time domain, where the compression ratio is called skimming ratio. The objective is to generate a "skimmed version" of the original video with high compression ratio, while preserving content information as much as possible. We proposed a novel hierarchical framework to progressively generate the summarized video. This framework takes into account the concepts of completeness, saliency, smoothness and ratio flexibility. The hierarchical

framework is shown to be effective with subjective tests. A possible future direction of this work is to exploit the semantic implications in concept primitives with supervised learning so as to generate higher-cognitive level shot groups for more sophisticated video structure representation.

For video compression, the goal is to minimize the bit rate for transmission or storage while maximizing the video quality. We introduced a two level resource allocation framework and conduct bit-allocation with respect to the region-of-interests on intra and inter frames. The future works may include using rate-quantization modeling and residue distortion estimation to achieve more accurate rate-control. For the proposed new intra prediction tool - transition based intra coding, our future work is to incorporate the current intra coding scheme in our content-aware framework.

REFERENCES

[1] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[2] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proceedings of IEEE Computer Vision and Pattern Recognization*, 2007.

[3] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform," in *Proceedings of IEEE Computer Vision and Pattern Recognization*, 2008.

[4] F. Liu and M. Gleicher, "Video retargeting: automating pan and scan," in *Proceedings of ACM Multimedia*, 2006.

[5] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Transactions on Graphics*, 2007.

[6] L. Wolf, M. Guttmann, and D. Cohen-Or, "Non-homogeneous content-driven video-retargeting," in *Proceedings of IEEE International Conference on Computer Vision*, 2007.

[7] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *Proceedings of IEEE Computer Vision and Pattern Recognization*, 2008.

[8] G. Hua, C. Zhang, Z. Liu, Z. Zhang, and Y. Shan, "Efficient scale-space spatiotemporal saliency tracking for distortion-free video retargeting," *Computer Vision*, pp. 182–192, 2010.

[9] T. Deselaers, P. Dreuw, and H. Ney, "Pan, zoom, scan – time-coherent, trained automatic video cropping," in *Proceedings of IEEE Computer Vision and Pattern Recognization*, 2008.

[10] L. Taycher, J.W. Fisher III, and T. Darrell, "Combining object and feature dynamics in probabilistic tracking," in *Proceedings of IEEE Computer Vision and Pattern Recognization*, 2005.

[11] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of IEEE Computer Vision and Pattern Recognization*, 1994.

[12] S. de Brouwer, M. Missal, G. Barnes, and P. Lefevre, "Quantitative analysis of catch-up saccades during sustained pursuit," *Journal of Neurophysiology*, vol. 87, no. 4, pp. 1772–1780, 2002.

[13] J.S. Boreczky and L.A. Rowe, "Comparison of video shot boundary detection techniques," *Journal of Electronic Imaging*, 1996.

[14] N.V. Patel and I.K. Sethi, "Video shot detection and characterization for video databases," *Pattern Recognition*, 1997.

[15] SJ Sangwine and TA Ell, "Hypercomplex Fourier transforms of color images," in *Proceedings of International Conference on Image Processing*, 2001.

[16] P.J. Rousseeuw and A.M. Leroy, *Robust regression and outlier detection*, Wiley-IEEE, 2003.

[17] Y. Li, T. Zhang, and D. Tretter, "An overview of video abstraction techniques," *HP Laboratories Palo Alto, Tech. Report No. HPL-2001-191*, 2001.

[18] H.S. Chang, S. Sull, and S.U. Lee, "Efficient video indexing scheme for content-based retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1269–1279, 1999.

[19] D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," in *Proceedings of ACM Multimedia*. ACM, 1998.

[20] A. Hanjalic and H.J. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1280–1289, 1999.

[21] Y. Gong and X. Liu, "Video summarization using singular value decomposition," in *Proceedings of IEEE Computer Vision and Pattern Recognization*, 2000.

[22] M. Mills, J. Cohen, and Y.Y. Wong, "A magnifier tool for video data," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1992.

[23] Y. Taniguchi, A. Akutsu, Y. Tonomura, and H. Hamada, "An intuitive and efficient access interface to real-time incoming video based on automatic indexing," in *Proceedings of ACM Multimedia*. ACM, 1995.

[24] K. Otsuji, Y. Tonomura, and Y. Ohba, "Video browsing using brightness data," in *Proceedings of SPIE*, 1991.

[25] N. Omoigui, L. He, A. Gupta, J. Grudin, and E. Sanocki, "Time-compression: systems concerns, usage, and benefits," in *Proceedings of ACM Conference on Computer Human Interaction*. ACM, 1999.

[26] A. Amir, D. Ponceleon, B. Blanchard, D. Petkovic, S. Srinivasan, and G. Cohen, "Using audio time scale modification for video browsing," in *Proceedings of Hawaii International Conference on System Sciences*, 2000.

[27] G.W. Heiman, R.J. Leo, and G. Leighbody, "Word intelligibility decrements and the comprehension of time-compressed speech," *Perception and Psychophysics*, vol. 40, no. 6, pp. 407–411, 1986.

[28] Y. Li, S.H. Lee, C.H. Yeh, and C.C.J. Kuo, "Techniques for movie content analysis and skimming," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 79, 2006.

[29] D. Dementhon and D. Doermann, "Video summarization by curve simplification," in *Proceedings of ACM Multimedia*, 1998, pp. 211–218.

[30] Y. Gong and X. Liu, "Video summarization using singular value decomposition," in *Proceedings of IEEE Computer Vision and Pattern Recognization*, 2000, pp. 174–180.

[31] M. Padmavathi, R. Yong, and et. al, "Keyframe-based video summarization using Delaunay clustering," *International Journal on Digital Libraries*, vol. 6, no. 2, pp. 219–232, 2006.

[32] Z. Cernekova and Ioannis Pitas, "Information theory-based shot cut/fade detection and video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 82–91, 2005.

[33] Yuxin Peng and Chong-Wah Ngo, "Clip-based similarity measure for query-dependent clip retrieval and video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 5, pp. 612–627.

[34] Zhu Li, Guido M. Schuster, and et. al, "MINMAX optimal video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, pp. 1245–1256, 2005.

[35] J. Nam and A.H. Tewfik, "Dynamic video summarization and visualization," in *Proceedings of ACM Multimedia*, 1999, pp. 53–56.

[36] N. Vasconcelos and A. Lippman, "A Bayesian framework for semantic content characterization," in *Proceedings of IEEE Computer Vision and Pattern Recognization*, 1998, pp. 566–571.

[37] S. Lu, M.R. Lyu, and I. King, "Semantic video summarization using mutual reinforcement principle and shot arrangement patterns," in *Multimedia Modelling Conference*, 2005, pp. 60–67.

[38] D.M. Russell, "A design pattern-based video summarization technique: moving from low-level signals to high-level structure," in *International Conference on System Sciences*, 2000.

[39] B.W. Chen, J.C. Wang, and J.F. Wang, "A novel video summarization based on mining the story-structure and semantic relations among concept entities," *IEEE Transactions on Multimedia*, vol. 11, no. 2, pp. 295–312, 2009.

[40] G. Evangelopoulos, A. Zlatintsi, G. Skoumas, K. Rapantzikos, A. Potamianos, P. Maragos, and Y. Avrithis, "Video event detection and summarization using audio, visual and text saliency," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*. IEEE Computer Society, 2009.

[41] Y.F. Ma, L. Lu, H.J. Zhang, and M. Li, "A user attention model for video summarization," in *Proceedings of ACM Multimedia*. ACM New York, NY, USA, 2002.

[42] Y.F. Ma, X.S. Hua, L. Lu, and H.J. Zhang, "A generic framework of user attention model and its application in video summarization," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 907–919, 2005.

[43] C.W. Ngo, Y.F. Ma, and H.J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 296–305, 2005.

[44] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," *CUIDADO IST Project Report*, pp. 1–25, 2004.

[45] D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," *Machine Learning*, pp. 4–15, 1998.

[46] D.G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision*, 1999.

[47] J. Kim, H Chang, K. Kang, M. Kim, and H. Kim, "Summarization of news video and its description for content-based access," *International Journal on Image System and Technology*, vol. 13, no. 5, pp. 267–274, 2004.

[48] SG Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[49] S. Chu, S. Narayanan, and J. Kuo, "Environmental sound recognition using MP-based features," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008.

[50] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[51] T. Syeda-Mahmood and D. Ponceleon, "Learning video browsing behavior and its application in the generation of video previews," in *Proceedings of ACM Multimedia*. ACM, 2001.

[52] Y. Gao and Q. Dai, "Shot-based similarity measure for content-based video summarization," in *Proceedings of International Conference on Image Processing*, 2008.

[53] "Muscle movie database v3.0," http://poseidon.csd.auth.gr/EN/MUSCLE_moviedb, 2007.

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

[55] L.S. Karlsson and M. Sjostrom, "Improved ROI video coding using variable Gaussian pre-filters and variance in intensity," in *Proceedings of International Conference on Image Processing*, 2005, vol. 2.

[56] C.W. Lin, Y.C. Chen, and M.T. Sun, "Dynamic region of interest transcoding for multipoint video conferencing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 10, pp. 982–992, 2003.

[57] M.J. Chen, M.C. Chi, C.T. Hsu, and J.W. Chen, "ROI video coding based on H. 263+ with robust skin-color detection technique," in *Proceedings of IEEE International Conference on Consumer Electronics*, 2003.

[58] H. Wang and S. Kwong, "Rate-distortion optimization of rate control for H. 264 with adaptive initial quantization parameter determination," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 140–144, 2008.

[59] D. Chai, KN Ngan, and A. Bouzerdoum, "Foreground/background bit allocation for region-of-interest coding," in *Proceedings of IEEE International Conference on Image Processing*, 2000, vol. 2, pp. 923–926.

[60] Y. Liu, Z.G. Li, and Y.C. Soh, "Region-of-interest based resource allocation for conversational video communication of H. 264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 134–139, 2008.

[61] L.A. Vese and T.F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *International Journal of Computer Vision*, vol. 50, no. 3, pp. 271–293, 2002.

[62] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40, no. 10-12, pp. 1489–1506, 2000.

[63] Joint Video Team, "Draft of version 4 of H.264/AVC," January 2005.

[64] JCT-VC, "Description of video coding technology proposal by Tandberg, Nokia, Ericsson," April 2010.

[65] T. Shiodera A. Tanizawa and T. Chujoh, "Bidirectional intra prediction (VCEG-AE14)," in *31rd VCEG Meeting*, 2007.

[66] A. Tanizawa T. Shiodera and T. Chujoh, "Simulation results of bidirectional intra prediction on KTA software version 1.3(VCEG-AF06)," in *32rd VCEG Meeting*, 2007.

[67] T. Chujoh T. Shiodera and A. Tanizawa, "Improvement of bidirectional intra prediction (VCEG-AG08)," in *33rd VCEG Meeting*, 2007.

[68] "Key technology area reference software," http://iphome.hhi.de/suehring/tml/download/KTA/, 2002.

[69] W. Zeng and B. Liu, "Geometric-structure-based error concealment with novel applicationsin block-based low-bit-rate coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 648–665, 1999.

[70] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 1, pp. 13–18, 1979.

[71] VR Algazi, GE Ford, and R. Potharlanka, "Directional interpolation of images based on visual properties and rank order filtering," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1991, pp. 3005–3008.

[72] K.H. Jung, J.H. Chang, and C.W. Lee, "Error concealment technique using projection data for block-based image coding," in *Proceedings of SPIE*, 1994, vol. 2308, p. 1466.

[73] Sencore, "CMA 1820 compressed media analyzer software," 2008.

[74] Elecard, "Elecard stream analyzer v 2.0 user guide," 2008.

[75] ISO/IEC, "Draft call for evidence on high-performance video coding (HVC), MPEG document N10363," January 2009.

[76] Toshiba Corporation, "Quadtree-based adaptive loop filter," January 2009.

BIOGRAPHICAL SKETCH

Taoran Lu received her bachelor's degree in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006. She received her master's degree at the Department of Electrical and Computer Engineering of the University of Florida in 2007. In December 2010, she received her Ph.D. from the University of Florida. During the summer of 2009 and 2010, she worked as a research intern in Thomson Corporate Research (Technicolor Research and Innovation) at Princeton, New Jersey. Her research interests include advanced video coding (H.264, KTA), video content modeling, video streaming, video processing on video summarization and adaptation, region-of-interest detection and saliency analysis.