THE UNIVERSITY OF
SYDNEY

COPYRIGHT AND USE OF THIS THESIS

This thesis must be used in accordance with the provisions of the Copyright Act 1968.

Reproduction of material protected by copyright may be an infringement of copyright and copyright owners may be entitled to take legal action against persons who infringe their copyright.

Section 51 (2) of the Copyright Act permits an authorized officer of a university library or archives to provide a copy (by communication or otherwise) of an unpublished thesis kept in the library or archives, to a person who satisfies the authorized officer that he or she requires the reproduction for the purposes of research or study.

The Copyright Act grants the creator of a work a number of moral rights, specifically the right of attribution, the right against false attribution and the right of integrity.

You may infringe the author's moral rights if you:

- fail to acknowledge the author of this thesis if you quote sections from the work
- attribute this thesis to another author
- subject this thesis to derogatory treatment which may prejudice the author's reputation

For further information contact the University's Director of Copyright Services

sydney.edu.au/copyright

NOVEL PERSPECTIVES AND APPROACHES TO VIDEO SUMMARIZATION

A thesis submitted in fulfilment of the requirements for the
degree of Doctor of Philosophy in the
School of Information Technologies
Faculty of Engineering & Information Technologies at
The University of Sydney

Genliang Guan

June 2015

© Copyright by Genliang Guan 2015
All Rights Reserved

Abstract

The increasing volume of videos and the mounting interest from consumers in accessing to such repositories require efficient and effective techniques to index and structure videos. Video summarization is such a technique that extracts the essential information from a video, so that tasks such as comprehension by users and video content analysis (e.g., indexing and classification) can be conducted more effectively and efficiently. The research presented in this thesis investigates three novel perspectives of the video summarization problem and provides approaches to such perspectives.

Our first novel perspective is to employ local keypoint to perform keyframe selection. While majority of the existing methods use global visual features to characterize each video frame, where local visual details are neglected, a local keypoint based framework is proposed for the first time to tackle this issue. Two criteria, namely Coverage and Redundancy, are introduced to guide the keyframe selection process in order to identify those which represent maximum video content and share minimum redundancy. Experiments presented in this thesis indicate that this approach achieves better performance than some state-of-the-art video summarization techniques. In order to more effectively and efficiently deal with long videos, a top-down strategy is proposed, which splits the summarization problem to two sub-problems: scene identification and scene summarization. In the first step, each frame is characterized with global visual features and a scalable clustering method is utilized to group frames into scenes. Secondly, local visual features are used to find the representative keyframes

within each scene. Superior results are obtained from experiments over two publicly available datasets.

Our second perspective is to formulate the task of video summarization to the problem of sparse dictionary reconstruction. In other words, the task is to best reconstruct the original video sequence with as few selected keyframes as possible. Different with the recently proposed sparse dictionary selection based method, our proposed method utilizes the true sparse constraint L_0 norm, instead of the relaxed constraint $L_{2,1}$ norm, such that keyframes are directly selected as a sparse dictionary that can well reconstruct all the video frames. An on-line version is further developed owing to the real-time efficiency of the proposed Minimum Sparse Reconstruction (MSR) principle. In addition, a Percentage Of Reconstruction (POR) criterion is proposed to intuitively guide users in selecting an appropriate length of the summary. Experimental results on two benchmark datasets with various types of videos demonstrate that the proposed methods outperform the sate-of-the-art approaches. In addition, an $L_{2,0}$ constrained sparse dictionary selection model is also proposed to further verify the effectiveness of sparse dictionary reconstruction for video summarization.

Lastly, we further investigate the multi-modal perspective of multimedia content summarization and enrichment. Video content, or any media content in general, do not only contain visual information, but textual and audio information as well. Therefore, the future of multimedia content summarization, as we believe, will be multi-modal content analysis and summarization. Visual modality plays a crucial role in our daily activities such as comprehending information and acquiring knowledge. Meanwhile, there are abundant images and videos on the Web. As a result, it is highly desirable to effectively organize such resources for content enrichment so as to facilitate comprehension and to improve user experiences in consuming textual stories such as news articles, documentaries, biographies, and Wikipedia entries, where limited visual aids

are provided. Therefore, we propose to address such an issue by utilizing abundant web resources to bridge the gap between these two modalities. Although there exist some studies on organizing web images for very specific concepts such as objects (e.g., car and table), persons, and landmarks, little research has been conducted for textual information at the story level. Our work, namely *StoryImaging*, is one step further than the traditional usage of image search systems. With the support of web scale images, our proposed approach is capable of enriching arbitrary textual stories with visual content.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

First and foremost, I would love to express my gratitude and thanks to my advisors, Dr. Zhiyong Wang and Prof. Dagan Feng. I have learned a lot from them as a researcher and also as a person par excellence. This thesis would not have been completed without their guidance and encouragement.

I would also like to thank my friends and colleagues in the School of Information Technologies, especially those from the Biomedical & Multimedia Information Technology (BMIT) Research Group. Their support has been invaluable throughout my PhD study, making my time both memorable and enjoyable. Another special thanks go to Dr. Shaohui Mei at Northwestern Polytechnical University (NWPU) in China. His insights and help support my research at a great magnitude.

My research is mainly supported by University of Sydney International Scholarship (also known as USydIS). Some of the studies presented in this thesis are partially supported by ARC (Australian Research Council) grants, National ICT Australia (NICTA), Natural Science Foundation of Shanxi Province Grant (No. 2010JZ011, 2013JQ801), National Natural Science Foundation of China (61201324, 61171154, 61371089), and NWPU foundation for fundamental research.

Last but not least, I thank my family for their unconditional love and support without which this journey would not have been possible.

THIS PAGE INTENTIONALLY LEFT BLANK

Publications

Journal Publications

[1] Shaohui Mei, Genliang Guan, Zhiyong Wang, Shuai Wan, Mingyi He, David Dagan Feng. Video Summarization via Minimum Sparse Reconstruction. *Pattern Recognition*, 48(2):522-533, 2015.
Chapter 5 is based on this publication.

[2] Genliang Guan, Zhiyong Wang, Shaohui Mei, Max Ott, Mingyi He, David Dagan Feng. A Top-down Approach for Video Summarization. *ACM Transactions on Multimedia Computing, Communications and Applications*, 11(1): Article No. 4, 2014.
Chapter 4 is based on this publication.

[3] Shiyang Lu, Zhiyong Wang, Tao Mei, Genliang Guan, David Dagen Feng. A Bag-of-Importance Model with Locality-Constrained Coding based Feature Learning for Video Summarization. *IEEE Transactions on Multimedia*, 16(6):1497-1509, 2014.

[4] Genliang Guan, Zhiyong Wang, Shiyang Lu, Jeremiah Da Deng, David Dagan Feng. Keypoint-Based Keyframe Selection. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4):729-734, 2013.
Chapter 3 is based on this publication.

[5] Zhiyong Wang, Genliang Guan, Yu Qiu, Li Zhuo, David Dagan Feng. Semantic context based refinement for news video annotation. *Multimedia Tools and Applications*, 67(3):607-627, 2013.

Conference Publications

[1] Shaohui Mei, Genliang Guan, Zhiyong Wang, Mingyi He, Shuai Wan, and David Dagan Feng. Iterative Keyframe Selection By Orthogonal Subspace Projection. In *IEEE International Conference on Image Processing*, Pages 2874-2878, 2014.

[2] Shaohui Mei, Genliang Guan, Zhiyong Wang, Mingyi He, Xian-Sheng Hua, and David Dagan Feng. $L_{2,0}$ constrained Sparse Dictionary Selection for Video Summarization. In *IEEE International Conference on Multimedia & Expo*, Pages 1-6, 2014. Chapter 6 is based on this publication.

[3] Genliang Guan, Zhiyong Wang, Kaimin Yu, Shaohui Mei, Mingyi He, David Dagan Feng. Video Summarization with Global and Local Features. In *IEEE International Conference on Multimedia & Expo Workshops*, Pages 570-575, 2012.

[4] Gang Tian, Genliang Guan, Zhiyong Wang, David Dagan Feng: What is happening: annotating images with verbs. In *ACM international conference on Multimedia*, Pages 1077-1080, 2012.

[5] Kaimin Yu, Zhiyong Wang, Genliang Guan, Qiuxia Wu, Zheru Chi, David Dagan Feng: How Many Frames Does Facial Expression Recognition Require? In *IEEE International Conference on Multimedia & Expo Workshops*, Pages 290-295, 2012.

[6] Kaimin Yu, Zhe Li, Genliang Guan, Zhiyong Wang, David David Feng: Unsupervised text segmentation using LDA and MCMC. In *Australasian Data Mining Conference*, Pages 21-26, 2012.

[7] Genliang Guan, Zhiyong Wang, Xian-Sheng Hua, David Dagan Feng: StoryImaging: a media-rich presentation system for textual stories. In *ACM International Conference on Multimedia*, Pages 775-776, 2011.
Chapter 7 is based on this publication.

[8] Yu Qiu, Genliang Guan, Zhiyong Wang, David Dagan Feng: Improving News Video Annotation with Semantic Context In *Digital Image Computing: Techniques and Applications*, Pages 214-219, 2010.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis explores the task of video summarization from three novel perspectives and provides approaches and experimental findings in these perspectives. This chapter introduces the motivations behind our work and its objectives. Then the exploration of our novel approaches to video summarization is discussed. Finally, this thesis is concluded with a discussion of the major contributions and an outline of the structure of the thesis.

1.1 Motivations

Video is becoming a more and more prominent part of our daily life. TV programs and news videos are widely watched, recorded, and shared on social networks (e.g., Youtube[1], Vimeo[2] and Vine[3]). As reported by Youtube Statistics 2014 [1], there were 100 hours of videos uploaded to Youtube every minute. That is, about 16 years long videos were produced in just one day. With the ever-increasing number of videos we produce and consume, how to manage such enormous visual content is emerging as a

[1] http://www.youtube.com
[2] http://vimeo.com
[3] https://vine.co/

challenging topic in the research area. Video summarization, as being one of the hot tasks in this topic, aims to make people consume video more effectively by identifying the most important information.

Early research on this task started at late 1990s [2][3], and it has attracted more and more attentions since then with the popularity of video consumption and production. Because of the popular use of of global visual features (e.g., texture and color) in visual data analysis in this period, most of the existing approaches to video summarization utilize only the global visual features of each video frame to identify important keyframes. Consequently, subtle details within video frames are not taken into account in video summarization. On the other hand, local visual features, e.g. the scale-invariant feature transform (SIFT) descriptor [4] and its variants, have been showing distinctive representation power in several tasks recently, including object recognition and image classification.

Apart from the consideration of local visual features, video summarization can also be treated as a problem of data reconstruction, where the goal is to select a number of keyframes to reconstruct or represent the original video with minimum error and redundancy. This perspective is inspired by a special type feature learning called sparse coding. It is a class of unsupervised methods for learning over-complete basis vectors to represent the original feature vectors efficiently as a linear combination of these basis vectors. There exist some approaches to video summarization using sparse dictionary selection, however this is a new direction that is worth more in-depth research and exploration.

Last but not least, video summarization can be regarded as a subset of multimedia content summarization. Multi-modal content analysis will become the next hot topic, which integrates the information of textual, audio and visual features of a piece of content to generate a more meaningful and content-rich presentation. This is especially

beneficial to the presentation of a large piece of content (e.g., a video or a text story) to users. Most of the current efforts still focus on summarizing content of individual modalities, and pioneering research on this perspective is limited.

1.2 Objectives

Video summarization is not a new research topic, but there are many new perspectives of this task awaiting to be tested and explored further. The goal of this thesis is not to harvest all possible directions and solutions to this problem, but to provide research findings on some noteworthy perspectives and inspire related future works. Towards this goal, the objectives of this thesis can be split into three parts which will be pursued separately.

Our first objective is to verify the effectiveness of using local visual features in the process of keyframe selection. Since previous works lack the use of local visual features which have been prominently used in object recognition and image classification only, it would be interesting to find out how this feature could help identify important keyframes. Furthermore, the huge time cost of local feature detection and keypoint matching is a non-negligible factor that affects the effectiveness and efficiency of such approach, so this issue should also be addressed.

Our second objective is to investigate the reformulation of the video summarization problem into a mathematical problem of data reconstruction. While there are many ways to identify a frame as an important keyframe, from simple shot detection and clustering to complex graph cutting algorithms, formulating it as a mathematical problem with established theories could easily lead to promising results and further improvements.

Our last objective is to have a futuristic preview of multimedia summarization. We

are already in the age of digitization, and many digital content already contains multi-modal information. Summarizing multimedia content or enriching single-modal content with multi-modal information will become more and more common in the future. We have taken an experimental attempt to tackle this task in this thesis.

1.3 Contributions

The main contributions of this thesis can be summarized as follows:

- In our first perspective, a local visual feature based method is proposed to tackle the problem of video summarization. To the best of our knowledge, this is one of the first attempts in the world on constructing a unique keypoint pool for video summarization. At first, local keypoints with distinctive features are extracted from every video frame. Then, all unique keypoints from all frames are merged via keypoint matching to become a global pool. Lastly, important keyframes are identified to achieve maximum coverage of the global pool and minimum redundancy among the selected keyframes.

- To further enhance the aforementioned approach, a top-down approach is proposed to split the video summarization problem into scene identification and scene summarization, where global and local visual features are exploited. Scene identification is achieved by grouping similar frames into scenes, and scene summarization is formalized as the aforementioned keypoint pool coverage problem. Additionally, the efficiency of this problem is improved by a kd-tree forest based local visual word model.

- In our second perspective, a Minimum Sparse Reconstruction (MSR) based approach is formulated by utilizing a selection matrix, such that video summarization is achieved by utilizing minimum number of keyframes to reconstruct the

entire video as accurate as possible. An L_0 norm based constraint is imposed to ensure true sparsity such that keyframes are selected directly according to the selection matrix. Two efficient and effective MSR based algorithms are proposed for off-line and on-line applications, respectively. Furthermore, a scalable strategy is designed to provide flexibility for practical applications. The proposed percentage of reconstruction (POR) criterion can be tuned to extract a keyframe set at different levels of reconstruction of the original video sequence. On the other hand, an $L_{2,0}$ based sparse dictionary selection model is also proposed as another method of sparse dictionary reconstruction. A simultaneous orthogonal matching pursuit (SOMP) based keyframe extraction algorithm is proposed to obtain an approximate solution for the proposed $L_{2,0}$ based problem directly without smoothing the penalty function. Thus, the contribution of non-keyframes for reconstruction is eliminated by strictly confining the reconstruction coefficients of non-keyframes to zero.

- In our third and last perspective, we conduct a pioneering attempt to perform multi-modal content summarization. A text to image system, so called *Story-Imaging*, is implemented to automatically enrich a given textual story by informatively organizing relevant web images. To the best of our knowledge, this work is one of the first efforts harvesting web images for text story illustration.

1.4 Organization of The Thesis

The remainder of this thesis is organized as follows.

Chapter 2 presents a comprehensive review of the state-of-the-art in video summarization categorized by different features being utilized, followed by the discussion of datasets being used in the current works.

Chapter 3 presents a novel keyframe selection method based on local keypoints. It firstly introduces the keypoint based video shot representation, and explains how a keypoint pool is constructed for the whole video. Then it gives details of a greedy algorithm tailored to efficiently solve the keyframe selection problem, followed by comprehensive experiments with case study examples and quantitative evaluation demonstrating the effectiveness of this method. Computational complexity is also discussed, the weakness of high computational cost is also mentioned, which will be addressed in the following chapter.

Chapter 4 elaborates a top-down framework to address the weakness of the method presented in Chapter 3 and to improve the effectiveness for long videos. X-means based scene identification with global visual features is firstly explained, and then the key steps of the keypoint feature based keyframe selection method to summarize each scene are discussed. Experiment section provides thorough evaluation, including the impact of different parameters, performance evaluation, case studies and video summaries with different lengths.

Chapter 5 introduces a novel video summarization technique based on Minimum Sparse Reconstruction. The MSR model is mathematically formulated, and solutions to this model are given to devise both an off-line and an on-line algorithms. Experimental results on two benchmark datasets are reported.

Furthermore, $L_{2,0}$ constrained sparse dictionary selection is also discussed in Chapter 6 as our another attempt to data reconstruction based video summarization. Simultaneous orthogonal matching pursuit (SOMP) based keyframe extraction algorithm is given in details for an approximate solution to this method.

Chapter 7 focuses on our pioneering work on multimedia summarization. A novel framework is presented to automatically illustrate a given textual story by informatively organizing relevant web images. We then showcase a simple yet effective and efficient

StoryImaging system with implementation details.

Finally, Chapter 8 summarizes our work and key findings, and provides some potential directions for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Literature Review

This chapter reviews the state-of-the-art in keyframe selection and video summarization, with particular attention to the works relevant to our own investigations. Video summarization has been well researched for decades [5] [6] [7]. The input and output of this task is simply illustrated in Figure 2.1. Given a sequence of frames from a target video, the task is to intelligently identify the frames or video segments of the highest importance as a summary.

In [3], Truong and Venkatesh categorized existing video summarization approaches into two types in terms of the forms of video summaries: keyframe based approaches and video skim based approaches. Keyframe based approaches select individual frames as a summary, while skim based approaches output a number of video segments as a summary, and our work belongs to the former category. As pointed out in that review, although video skims and keyframes are often generated in different ways, these two types of video summary can be easily converted from each other. Because of our research focus and the larger volume of existing works on keyframe based methods, our review will mainly focus on keyframe based video summarization.

Another review [8] by Money *et al.* divides existing research into three categories with respect to what kind of video characteristics are employed: internal (i.e., video

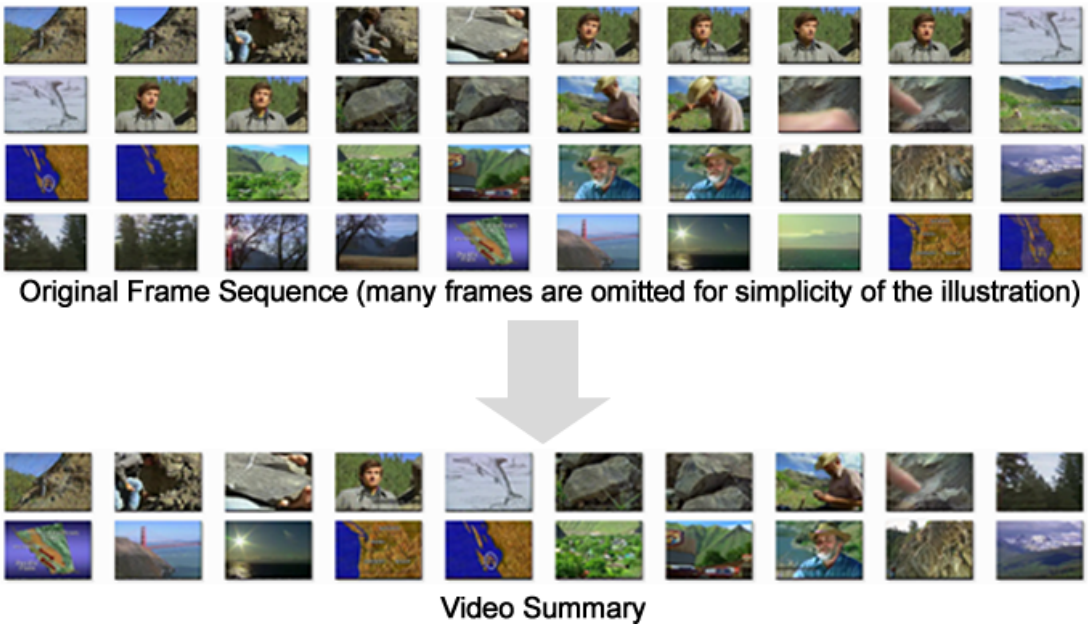


Figure 2.1: Illustration of Input and Output of Video Summarization.

signals), external (e.g., audio and subtitles) and hybrid (i.e., a combination of internal and external information).

In this chapter, at first, several visual feature based video summarization approaches are reviewed because they are more related to our research focus. They are categorized into two groups, i.e. global perspective and local perspective, based on the context in which each video frame is ranked and selected. Then a comprehensive review of other types of methods is provided based on semantic features and multi-modal features. Lastly, the datasets and evaluation metrics used in the literature and our experiments are discussed.

2.1 Visual Feature based Approaches

Most of existing works use visual feature based methods, which analyze the visual content of each video frame and identify the important ones as the summary. We are here categorizing those methods into two categories, namely global perspective based

and local perspective based approaches.

2.1.1 Global Perspective based Approaches

A video is a mixture of many similar and/or dissimilar frames. To select the most important keyframes representing a video, global perspective based approaches commonly make use of different clustering strategies. In [5], one of the first works in this category, Zhuang *et al.* proposed to utilize unsupervised clustering for keyframe selection. The proposed algorithm is both computationally simple and capable to adapt to visual content. Similarly, Avila *et al.* [9] and Furini *et al.* [10] also used k-means as the base of their method. Mundur *et al.* [11] represented each video frame as a multi-dimensional point in a complex graph, and Delaunay Triangulation is employed to cluster them. Besiris *et al.* [12] utilized graph connectivity and dominant set clustering to select keyframes by exploiting the connectivity information of prototype frames. The connectivity information for the prototypes is obtained from the whole set of data to improve video representation and reveal its structure. Ngo *et al.* [13] proposed a unified approach for video summarization based on the analysis of video structures and video highlights. Complete undirected graph is generated to represent a video, and the graph is partitioned into connected video clusters (a temporal graph) using normalized cut algorithm. Overall, simple clustering based methods basically identify keyframes from cluster centers, and different clustering algorithms can be applied.

On the other hand, keyframe selection has also been considered as an optimization problem where the original video should be optimally reconstructed with the selected keyframes as much as possible. In [14] by Gong *et al.*, Singular Value Decomposition was employed to project a raw frame-feature matrix into a lower dimensional space, and the set of keyframes was obtained by clustering . This method defines a metric to measure the amount of visual content contained in each frame cluster using its degree

of visual changes. The most static frame cluster was then identified, and the context value computed from it is used as the threshold to cluster the rest of the frames. Zhu *et al.* [15] formulated the keyframe identification problem as a temporal rate-distortion MINMAX optimization problem. Both an optimal dynamic programming based solution and a near-optimal Distortion Constrained Skip based solution is presented. By adopting Maximal Marginal Relevance (MMR) which is classically defined for text summarization [7], Li *et al.* derived a technique called Video Maximal Marginal Relevance (Video-MMR) for video summarization [16]. In MMR, text relevance is calculated based on its similarity to a specific text query, while in Video-MMR, keyframe coverage is calculated as the amount of similarity against the target video as a whole. Comparison with traditional k-means clustering algorithm supports its advantages.

2.1.2 Local Perspective based Approaches

A video is a sequence of consecutive frames. Considering that the final selected keyframes should be visually representative among their neighboring frames, local perspective based approaches focus on measuring visual redundancy in a temporal window that moves through the whole video. In [6] proposed by DeMenthon *et al.*, a high dimensional feature space was formed from global visual features on each frame, and all frames combined were treated as a trajectory curve in that space. This curve was simplified to lower dimension by a derived version of planar curve splitting algorithm, and eventually became a tree structure, where each level could be regarded as final keyframes. Similarly, three iso-content principles (i.e., Iso-Content Distance, Iso-Content Error and Iso-Content Distortion) were proposed in [17] to split the frame trajectory in a high dimensional feature space. Keyframes were selected at each partitioning locations, so they are equal-distant with regards to the principle in use. Recently, Cong *et al.* [18] proposed to project each frame (represented with a global feature) into a sparse feature

space, and a dictionary of key frames is selected such that the original video can be best reconstructed from this representative dictionary. The selected keyframes are those corresponding to the local maxima of the weight curve formed by sparse representation of each frame.

2.2 Semantics and Structure based Approaches

Many approaches also focus on bringing mid/high level semantics and video structure into video summarization, such as domain specific semantics [19] [20], 4W concepts (Who, What, Where, When) [21], events [22][23], camera motion [24] and human attentions [25][26]. In [27][28], textual semantics context is also used to enhance video annotation.

The work proposed by Vasconcelos *et al.* [19] is a very early attempt to bring semantics into the structure of videos. They realized the fact that video production normally follows specific rules and conventions, which introduce some structures to the result video. A probabilistic Bayesian network was employed to capture the content structure, and a map was then generated between semantic and image features. However, this method requires a large amount of domain specific training data, and also depends on the fact that their learned model cannot be applied to unseen videos, which is not usually the case in the fast-changing industry and daily life videos. The work of Ekin *et al.* [20] places its focus on soccer videos only, which relies on domain-specific algorithms for goal detection, referee detection, and penalty-box detection, which are able to capture the highlights of a soccer event.

Chen *et al.* [21] presented a structural video content browsing system which provides users with a concept-organized and systematic view by integrating visual and text features and constructing a relational graph using four concept entities, i.e. "who," "what," "where," and "when". Therefore, not only the browsing efficiency is enhanced,

but the user can also select what they are interested in. Graph entropy model was then exploited to detect important shots and relations, which became selectable items for the users.

Regarding event-driven query, Hong *et al.* [22] and Wang *et al.* [23] provided a mechanism to summarize video search results by mining and threading "key" shots, so that an overview of main content of these videos was generated for quick user consumption . These approaches deal with multiple videos in a prepared search results. Lu *et al.* [29] aimed to select a short chain of video sub shots to describe the essential information of the video. They used a random-walk based metric of influence between sub shots that reflects how visual objects contribute to the progression of events. Their summary can then captures how event are linked to each other rather than simple object co-occurrence.

In [24] proposed by Luo *et al.*, a psycho-visual study was firstly conducted to create an evaluation dataset and also to find out the criteria used by human judges so they can design the algorithm to better fit the criteria. They observed that spatio-temporal changes provide semantically meaningful information about scenes of the video and the general intents of the camera operator. As a result of this finding, a video clip was segmented into parts based on several types of camera motion (e.g., pan, zoom, pause, steady), and in each segment fine-tuned rules were employed to extract candidate keyframes.

2.3 Multi-modal Feature based Approaches

A number of studies also utilize multi-modal features (i.e., visual, audio and textual features) for video summarization.

Pan *et al.* [30] proposed a multi-modal story-oriented video summarization (MMSS) method which provides a domain-independent, graph-based framework. A general

framework is introduced to mine the cross-modal correlations among different modalities such as frames, terms and logos in video clips. The mined cross-modal correlations were then exploited for story summarization and video retrieval.

Ma *et al.* [25] indicated that human attention is an effective and efficient mechanism for information prioritizing and filtering. Based on that, a set of modeling methods for visual and aural attentions is proposed to better video understanding. They defined viewer attention through multiple sensory perceptions, i.e. visual and aural stimulus as well as partly semantic understanding, so as to identify keyframes based on importance ranking. Dong *et al.* [31] utilized both textual and visual information, and semantic concepts are labeled to each video segment. As a result, a feature space derived from the semantic concepts is exploited to measure the importance of each video segment.

Similarly, Evangelopoulos *et al.* [26] also formulated perceptual attention as salient events in the audio and visual streams. The presence of salient events was identified on this audiovisual curve by a few geometrical features such as local extrema, sharp transition points and level sets. They further extended their work to integrate textual saliency [32]. Aural saliency is calculated by signals that quantify multi-frequency waveform modulations. Visual saliency is calculated by a spatio-temporal attention model based on visual features such as brightness, color, and orientation. Furthermore, textual saliency is derived from the part-of-speech tags over the video subtitles. The individual saliency streams are combined in a multi-modal curve, and a bottom-up video summarization algorithm is then applied on this saliency curve to generate a summary.

2.4 Dataset and Evaluation

To evaluate the performance of a video summarization method, a video dataset is required. However, not only the methodologies vary, but the evaluation datasets being

used in existing works are very diverse. Because of the lack of a well-known and well-prepared dataset, many of the existing works resort to creating their own datasets for evaluation. For early works, the dataset used in evaluation are especially limited. In [5] only one action movie and one romantic movie were utilized to show the results. Likewise, Vasconcelos *et al.* [19] also subjectively evaluated their results using the promotional trailers of two movies. Zhu *et al.* [15] also only used one video (the famous Foreman video[1]) for case study.

A number of existing works aim to handle different types of videos. Gong *et al.* [14] used different types of video sequences: news reports, documentary, political debate, and live coverage of a breaking event, each of which lasts for from 5 to 30 minutes. In [13], two videos with sound tracks from the MPEG-7 video collection[2] and three home videos were adopted for evaluation. Each tested video has two associated summaries, one with 10% of the original video length and the other with 25% of the original length. Twenty students were invited to assign two scores ranging from 0 to 100 to summary results in terms of informativeness and enjoyability. Luo *et al.* [24] and Cong *et al.* [18] shared the same dataset, which consists of 18 QuickTime clips of VGA resolution and frame rates from 24 to 30 fps. These video were selected from a database of 3000+ video clips. Three human judges selected key frames for each video clip to form the ground truth of the dataset. Similarly, a dataset consisting of more than 250 real life video sequences is used in [17]. Most of these videos are from sport events. The coast sequence, the table tennis sequence, and the hall monitor sequence, which are the widely known MPEG test sequences, were also included in the dataset. In [16], the video dataset was collected from the website "wikio.fr" which aggregated news from many different sources. This website is already offline. Twenty one video sets were collected in total. Each has between 3 and 15 videos with durations from a few seconds

[1] https://www.youtube.com/watch?v=9-v8O1bv6A0

[2] http://mpeg.chiariglione.org/standards/mpeg-7

to above 10 minutes. The genres of the videos are varied, including news, advertisement and movie.

On the other hand, some other works focus on domain specific videos due to the requirement for domain knowledge and training data. In [20], only soccer TV recordings were used. Chen *et al.* [21] selected only documentary videos as our experimental data because of easier and more accurate retrieval of the 4W concepts. Eight genres of documentary videos were chosen in the dataset: buildings, ocean, wildlife, war, ancient history, space, crime science, and art. Some of these videos were obtained at the Open Video Project website[3]), and some were downloaded at the Discovery Channel, British Broadcasting Corporation (BBC), and National Geographic Channel documentaries. In [29], over 12 hours of daily activity video was taken by 23 unique camera wearers, and results were evaluated by comparing its quality with multiple baselines with 34 human subjects.

The most widely used dataset in the literature is from the Open Video Project, which has been used by [9], [11], [10], [12] and [21]. The Open Video Project aims to host and maintain a repository of digitized video content for the research communities. Those video data can be used to study a broad range of research topics, such as video segmentation, video summarization, face recognition, and multimedia retrieval. A test collection is also included to enable systems to be compared against the same dataset. There are more than 2100 videos collected in this dataset, whose genres include documentary, educational, ephemeral, historical, lecture, public service and others. There are still some missing types of videos though, such as actions and movies. The durations of videos are also diverse, ranging from less that 1 minute to more than 10 minutes. The variety of genres and durations is very useful for comprehensive testing of performance. When it comes to video summarization, ground truth keyframes must be prepared for quantitative evaluation. Luckily, Avila *et al.* [9] have invited 5 users to

[3]http://www.open-video.org

annotate 50 selected videos from the dataset, which then can be used by others to test the performance.

Our experiments mainly used the Open Video Project together with the ground truth data provided by [9] to perform evaluation. Details of the experiment settings and evaluations will be discussed in Section 4.4.1 and Section 4.4.2 of Chapter 4.

2.5 Summary

As discussed in this chapter, many approaches to video summarization have been proposed. The majority of existing works focus on visual feature analysis to identify keyframes, while semantics and structure based approaches are emerging to bring in more high-level video understanding. Multi-modal feature based approaches are growing as well because single-modal feature analysis are getting more mature. On the other hand, the dataset and evaluation methods used in the existing works are diverse, and there is no commonly-recognized dataset and evaluation method for experiments. This is a still a weakness in this research area. In light of the above observations, we explore some new perspectives to video summarization in the following chapters, and take the most-widely-used Open Video Project datasets to evaluate our approaches.

Chapter 3

Keypoint based Video Summarization

This chapter introduces a video summarization framework utilizing local keypoints on video frames. It is based on results and includes text that have been published in [33].

3.1 Introduction

The proliferation of video acquisition devices and the mounting interest of consumers in the access to video repositories have significantly boosted the demand for effective and efficient methods in retrieving and managing such multimedia data. A video is structurally composed of a number of stories, each story is depicted with a number of video shots, and each shot is essentially a sequence of images (i.e., frames) [34]. Due to the inherent temporal continuity of the consecutive frames within a video shot, there exists a great deal of redundant information among those frames. Therefore, selecting a set of frames to represent a video shot has been crucial for effective and efficient video content analysis.

Most of the existing works utilize global visual feature of each video frame to identify keyframes. These approaches are however mainly subject to the following limitations. First, these approaches highly rely on global features such as color, texture

and motion information, though adapting them to local features may be possible. As a result, the risk exists that local details of frames will be neglected, which makes the selected keyframes less representative, though global features coarsely represent visual characteristics of an image. Second, it is difficult to decide how many keyframes should be selected. For example, it is always challenging to set an appropriate threshold when two adjacent frames are compared. For the clustering based approaches, it is generally an open issue to set a reasonable number of clusters without prior knowledge. Moreover, different metrics are proposed to fulfill this task, however, there is no intuitive way to assess the quality of selected keyframes in terms of representativeness, redundancy and completeness.

Recently, local features such as the scale-invariant feature transform (SIFT) descriptor [4] have played a significant role in many application domains of visual content analysis such as object recognition and image classification due to their distinctive representation capacity. Hence, it would be beneficial to characterize each frame with local visual descriptors derived from the keypoints within the frame, and keyframe selection is to identify a number of frames whose keypoints are representative for the scene.

In light of the above observations, we propose a keypoint based keyframe selection framework summarized as follows. Firstly, keypoints are identified from each frame and descriptors are extracted for each keypoint. Secondly, a global pool of unique keypoints is formed to represent the whole video shot through keypoint matching. Finally, representative frames which best cover the global keypoint pool are chosen as keyframes. Two criteria, namely Coverage and Redundancy [35], are devised to ensure that each keyframe is selected to maximize the coverage of the keypoint pool and to minimize introducing redundant keypoints.

3.2 Keypoint based Video Shot Representation

3.2.1 Keypoint Matching

Lowe's SIFT descriptor [4] is utilized for keypoint extraction and representation, though many other local features [36] are also applicable to our approach. SIFT descriptor was proposed in [4] to perform reliable matching between different views of an object or scene. For each detected keypoint, there are three steps to calculate its SIFT descriptor. Firstly, the image gradient magnitudes and orientations are computed, sampled from a 16×16 region around the keypoint. Secondly, in order to eliminate the influence introduced by small changes in the position of the window, the magnitude of each sample point is weighted by a Gaussian weighting function. Thirdly, these samples are accumulated into orientation histograms summarizing the contents over 4×4 subregions. The length of each orientation vector corresponds to the sum of the gradient magnitudes near that direction of the region. Therefore, SIFT descriptor of each keypoint is a $4 \times 4 \times 8 = 128$ dimension feature vector (i.e., a 4×4 array of orientation histograms with 8 orientation bins in each). The 128-dimensional descriptor is invariant to image scale and rotation, which is robust for many applications such as object recognition and image stitching. It has be shown to be very effective in the domain of object recognition, image stitching, video tracking and so on.

Straightforward keypoint matching based on SIFT descriptors will result in many false matches. Lowe proposed to improve matching robustness by imposing ratio test criterion (i.e. the ratio of the nearest neighbor distance to the second nearest neighbor distance is greater than a given threshold) [4]. However, there still exist two challenging problems;

Firstly, the cost of keypoint matching between two target frames is high. To exhaustively match keypoints, we have to calculate the distance between every pair of

keypoints in both frames, which is computationally expensive. In order to relieve this problem and take advantage of the continuity among adjacent frames, we adopt a matching strategy that considers only those candidate keypoints within a certain radius R of the target keypoint. Meanwhile, false matching can also be reduced with such a constraint. Secondly, there are a number of false-positive matches, and as a result, the global pool of keypoints K would contain noisy keypoints. To filter these false matches, the RANdom Sample Consensus algorithm (RANSAC) [37] is iteratively invoked to detect sets of geometrically consistent keypoint matches. This process is repeated until no further large set of matches (e.g. five matches in a group) can be found.

3.2.2 Keypoint Pool Construction

In order to build a global pool K from all keypoints k_x in each frame f_i to represent the content of a video shot, ideally every two frames f_i and f_j (a pair) within the shot should go through keypoint matching. However, such a strategy is very costly. Utilizing the inherent nature of visual continuity among consecutive video frames, we propose an Inter-window Keypoint Chaining scheme to constrain the pairing within a temporal window of size W without losing the discriminative power of keypoint matching, as illustrated in Fig. 3.1. Hence, keypoints are only matched within a window and chained across multiple windows. When a keypoint k_1 in frame f_i is matched with another keypoint k_2 in frame f_j, and the same keypoint k_2 is matched with a third keypoint k_3 in frame f_m, satisfying $|i-j| <= W$ and $|m-j| <= W$, we link these matches into a chain, which would finally contribute to the same unique keypoint in the global pool K without matching keypoints between f_i and f_m. As shown in our recent study [38], the window size can be adaptively determined by calculating visual variations between consecutive frames in terms of distribution correlation.

On the other hand, it may occur that true keypoint matches are dropped during

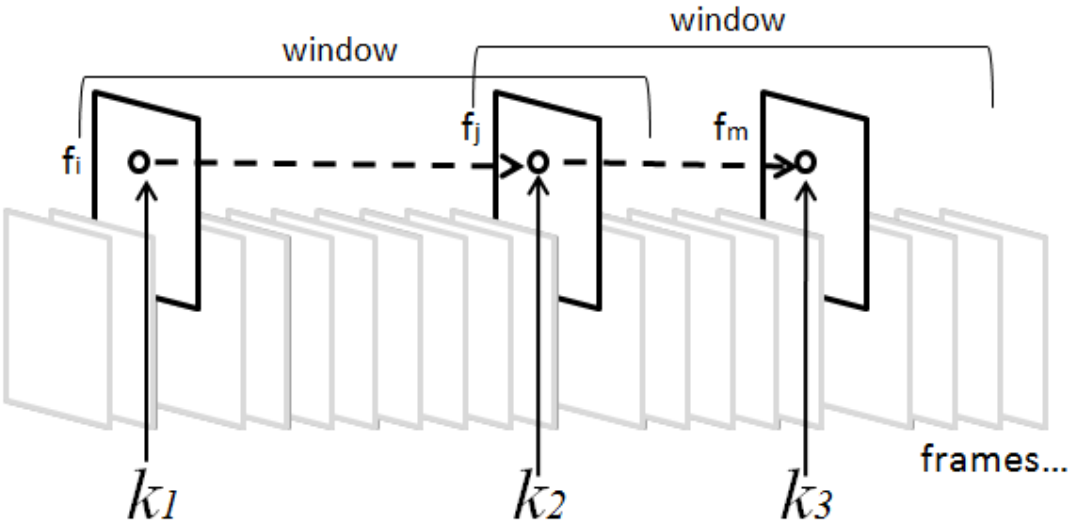


Figure 3.1: Illustration of Inter-window keypoint chaining with overlapped windows, where k_1, k_2, and k_3 are matched keypoints.

matching. In order to make the matching more reliable, we also propose Intra-Window Keypoint Chaining. As shown in Fig. 3.2, k_1 is matched with k_3 but not k_2, and k_2 is matched with k_3. In this case, k_1, k_2 and k_3 will also be linked by a single chain, which could ease the problem of missed matching (e.g. k_1 should be a true match with k_2).

After the keypoint chaining on frames, each keypoint either belongs to a chain of matched keypoints or becomes an singleton. All singleton keypoints, which are very likely to be noisy keypoints, are discarded. Each chain is represented by its HEAD keypoint and the number of keypoints on that chain, denoted by (k_x, N_x). The global keypoint pool K is then formed by aggregating all (k_x, N_x) (see Figure 3.3). As shown in Figure 3.4, each chain has different number of keypoints for a Tennis video shot. In order to reduce noisy chains, we further filter less important/unstable global keypoints by setting a threshold T for N_x.

To illustrate the usefulness of the keypoint pool, the number of keypoints of a video frame over time is shown in Figure 3.5 as an example, where the number of keypoints increases while the frame progresses temporally. As indicated by the horizontal dashed

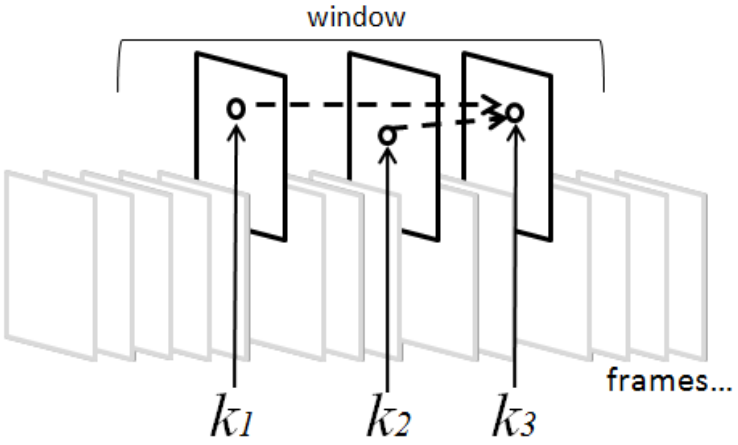


Figure 3.2: Illustration of Intra-window keypoint chaining within one window, where k_1, k_2, and k_3 are matched keypoints and merged into one chain.

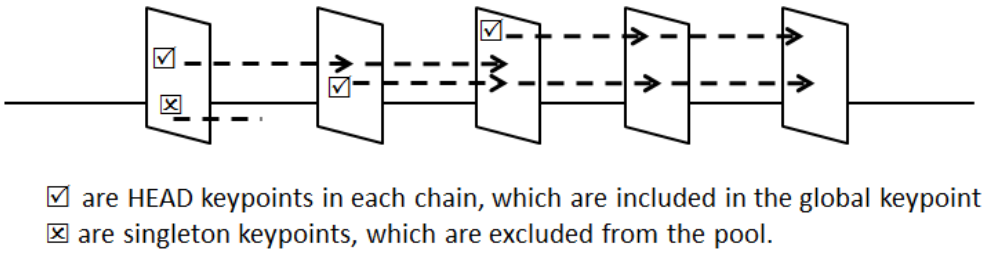


Figure 3.3: The global keypoint pool K is formed by all HEAD keypoints of each chain.

line, the keypoint pattern changes around the 35-th frame, which corresponds to the content change. Specifically in this example of the Tennis video, a small panning transition happens around the 35th frame, which could be detected from the frame-keypoint correspondence visualized in the figure.

3.3 Keyframe Selection

The goal of keyframe selection is to best represent a video shot with a minimal number of frames. That is, the keyframes are able to best represent the video shot while

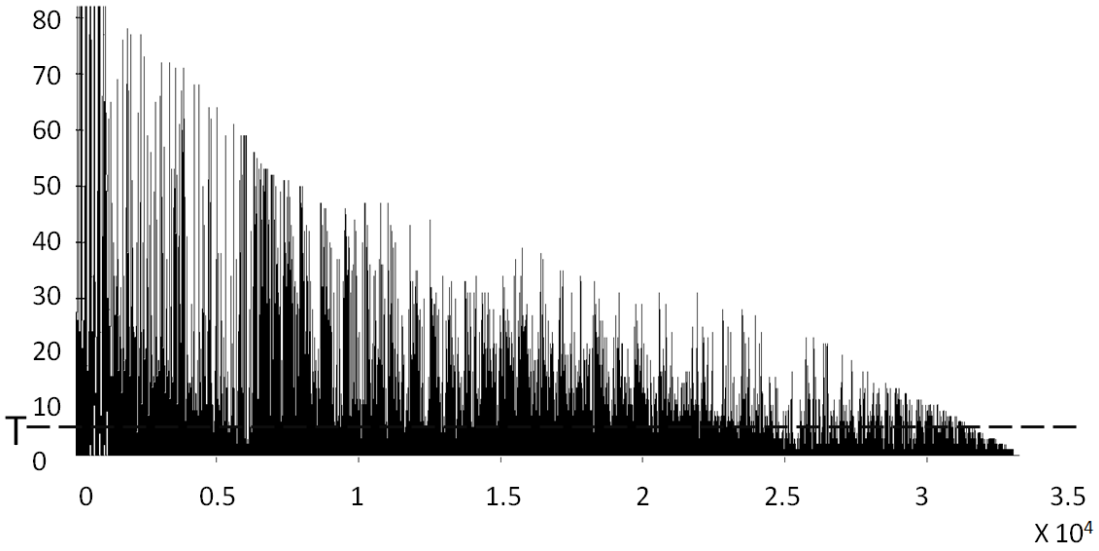


Figure 3.4: The number of keypoints (y-axis) along each chain (x-axis) for the Tennis video shown in Fig. 3.11.

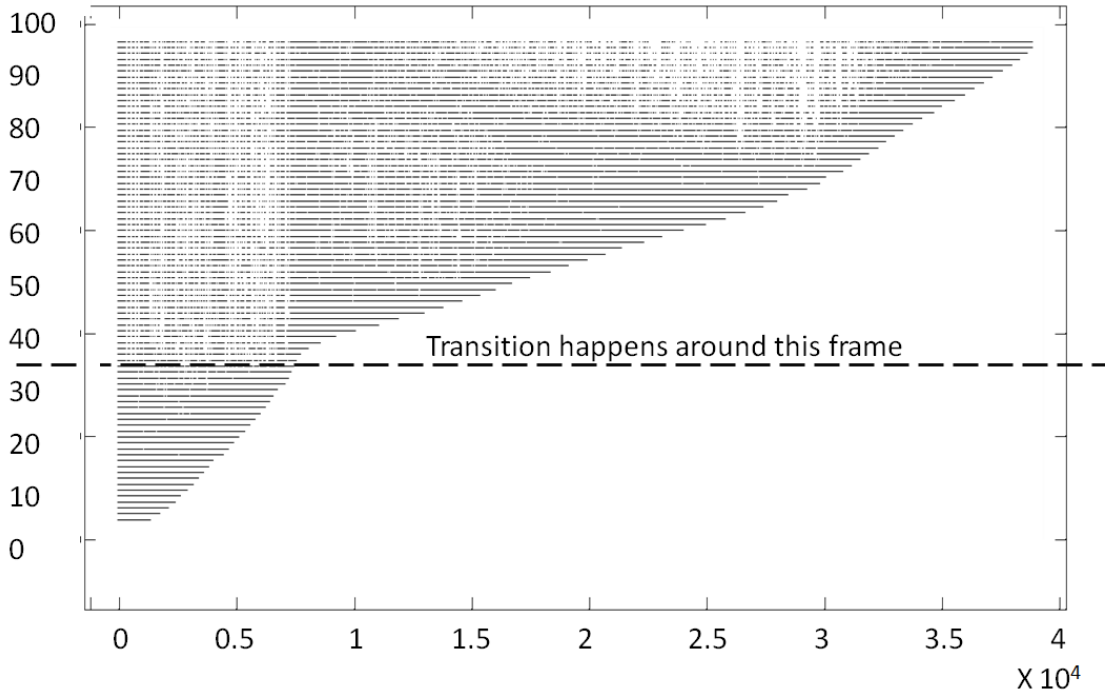


Figure 3.5: The number of keypoints over time for the Tennis video, where x-axis is the index of keypoints and y-axis is the index of frames. A dot denotes the existence of a keypoint in a frame.

minimizing redundancy among them. In our case, to ensure the best representation, the keypoints of those keyframes should cover the global keypoint pool as much as possible. Since this can be formulated as a variation of the well-known Set Cover Problem which has been proven to be NP-complete [39], we adopt a greedy algorithm to approximately tackle this issue. At first, we choose the frame with the highest number of keypoints against the keypoint pool. Then, at each iteration, a frame is chosen as a keyframe if it best helps improve the coverage while minimizing redundancy. Therefore, we devise two metrics, namely *Coverage* and *Redundancy*, to guide the selection process.

In the selection process, the pool is separated into two sets, $K_{covered}$ and $K_{uncovered}$. At the beginning of the process, $K_{uncovered}$ contains all keypoints in K and $K_{covered}$ is empty. For frame f_i, denote its keypoint set as FP_i, then the *Coverage* of the frame to the pool can be defined as the cardinality of the intersection between FP_i and the uncovered set:

$$C(f_i) = |FP_i \cap K_{uncovered}|. \tag{3.1}$$

Likewise, *Redundancy* is defined as how many keypoints it contains in $K_{covered}$, reflecting how redundant it is based on the covered content in the shot:

$$R(f_i) = |FP_i \cap K_{covered}|. \tag{3.2}$$

The influence of frame f_i at an iteration is calculated in (3.3) as a balance of $C(f_i)$ and $R(f_i)$ controlled by α.

$$Influence(f_i) = C(f_i) - \alpha R(f_i) \tag{3.3}$$

A simplified illustration of the calculation is presented in Fig. 3.6. In this example, f_1 has higher coverage but also higher redundancy than f_2, so f_2 will be favored during the selection.

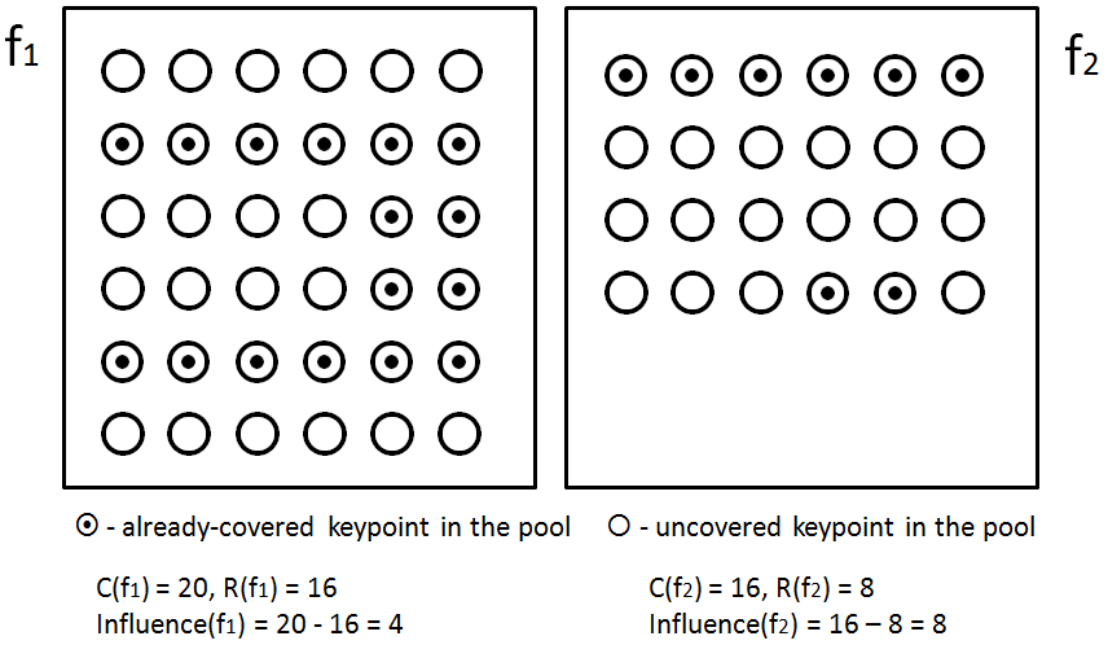


Figure 3.6: A toy sample of calculating of the influence of frames, where f_2 is selected because of its higher influence.

At the end of each iteration, the frame with the highest and positive influence value will be selected as a keyframe, and $K_{covered}$ and $K_{uncovered}$ will be updated based on the keypoints of the selected keyframe. The iteration repeats until all the keypoints are covered or a predefined coverage threshold of the pool K is reached.

3.4 Experiments and Discussions

3.4.1 Experimental Settings

We conduct experiments with two datasets. The first dataset is for case studies, consisting of 4 videos including the widely used Foreman and Coastguard videos and two TV news shots (Tennis video and Zooming video). The second dataset [1] is constructed from the Open Video Project (http://www.open-video.org) for quantitative evaluation.

[1] http://sydney.edu.au/engineering/it/ zhiyong/data/kfsyd.html

Table 3.1: The Testing Videos from the Open Video Project

Video Name	Start Frame	End Frame	# of Frames
v25 A New Horizon, segment 02	664	900	237
v28 A New Horizon, segment 05	3223	3440	218
v33 Take Pride in America, segment 03	540	650	111
v39 Senses And Sensitivity, Introduction to Lecture 4 presenter	1838	1934	97
v40 Exotic Terrane, segment 01	1790	1989	200
v49 America's New Frontier, segment 07	150	500	351
v57 Oceanfloor Legacy, segment 04	1600	1800	201
v58 Oceanfloor Legacy, segment 08	540	633	94
v63 Hurricane Force - A Coastal Perspective, segment 03	867	1012	146
v66 Drift Ice as a Geologic Agent, segment 05	766	977	212

As described in Table 3.1, it consists of 10 video shots across several genres (e.g. documentary, education, and history).

In our experiments, the results generally are not affected when the matching radius R is set above 100 and the window size W above 5. Hence we set the radius R to 100 (i.e., 100 pixels around a target keypoint) to reduce matching search space without sacrificing matching accuracy even in fast-motion scenes, and W to 5 so as to balance the computational cost and chaining accuracy. The threshold T to filter the unstable global keypoint affects the size of the keypoint pool and thus the granularity of details it captures. Empirically we have tried different settings in our experiments, but the results shown in the following section are based on $T = 5$ to reduce noisy keypoints without losing noticeable details.

Our approach (denoted as KBKS in the figures) is compared against three state-of-the-art approaches, Iso-Content Distance [17], Iso-Content Distortion [17] and Clustering [5]. For the first two approaches we use the same Color Layout Descriptor as

adopted in the original paper. For the clustering based method, we adopt the Colour and Edge Directivity Descriptor (CEDD) feature [40], which is a histogram representing color and texture features.

3.4.2 Case Studies

The sample frames for the four shots in discussion is presented in Fig. 3.8. The results for the Foreman video are displayed in Fig. 3.9. It is observed that our approach can capture different details when different coverage threshold values are specified. For example, the two frames under 73% coverage capture the key content, the foreman and the building. When the coverage is increased to 95%, different stages of the smiling face are captured. However, such details are missing in the results of the other methods, since they rely on global features. Meanwhile, it is also noticed that our approach misses the keyframe on the tower and sky. There are two reasons. One is that the transition is very short and some keypoint chains are discarded. The other is that there are not many keypoints due to a large portion of the uniform region and the influence score of those frames have been affected. In order to remedy this issue, we take global features into account by replacing (3.3) with the following equation:

$$InfluencNew(f_i) = \frac{C(f_i) - \alpha R(f_i)}{GolbalSim(f_i)}, \tag{3.4}$$

where

$$GolbalSim(f_i) = \sum_j Similarity(f_i, f_j). \tag{3.5}$$

That is, the influence of a frame f_i will be increased if it shares low similarity (i.e. small $GolbalSim(f_i)$) with other frames in terms of color and edge histogram. As shown in Fig. 3.7, such a simple strategy is able to effectively resolve the "missing sky" problem, though not being used in our other experiments.

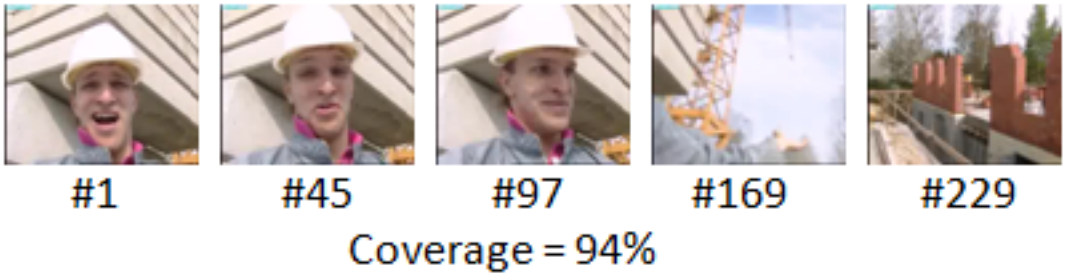


Figure 3.7: New keyframe selection results for the Foreman video.

Figure 3.8: Sample frames of the Foreman, Coastguard, Tennis, and Zooming videos (from top to down).

For the Coastguard video (See Fig. 3.10) capturing that one boat overtakes the other, our approach selects not only the frames with both boats, but also more frames to get a higher coverage of keypoints as the background of the boat (e.g., the building and trees) keeps changing. The other two methods do capture both boats, but do not reflect the background change very well. In addition, from our selected keyframes, it seems easier for audience to understand the overtaking process.

In Fig. 3.11, the Tennis video contains two actions of the player with a very short panning and fading transition in between. Our selection algorithm clearly identifies these two action frames with a high keypoint coverage of 97%. The clustering-based method achieves the similar result with the help of predefined the number clusters (i.e., 2), and the Equidistance method selects the first and last frames.

The last video is a short zoom-out footage. Our approach selects one keyframe near

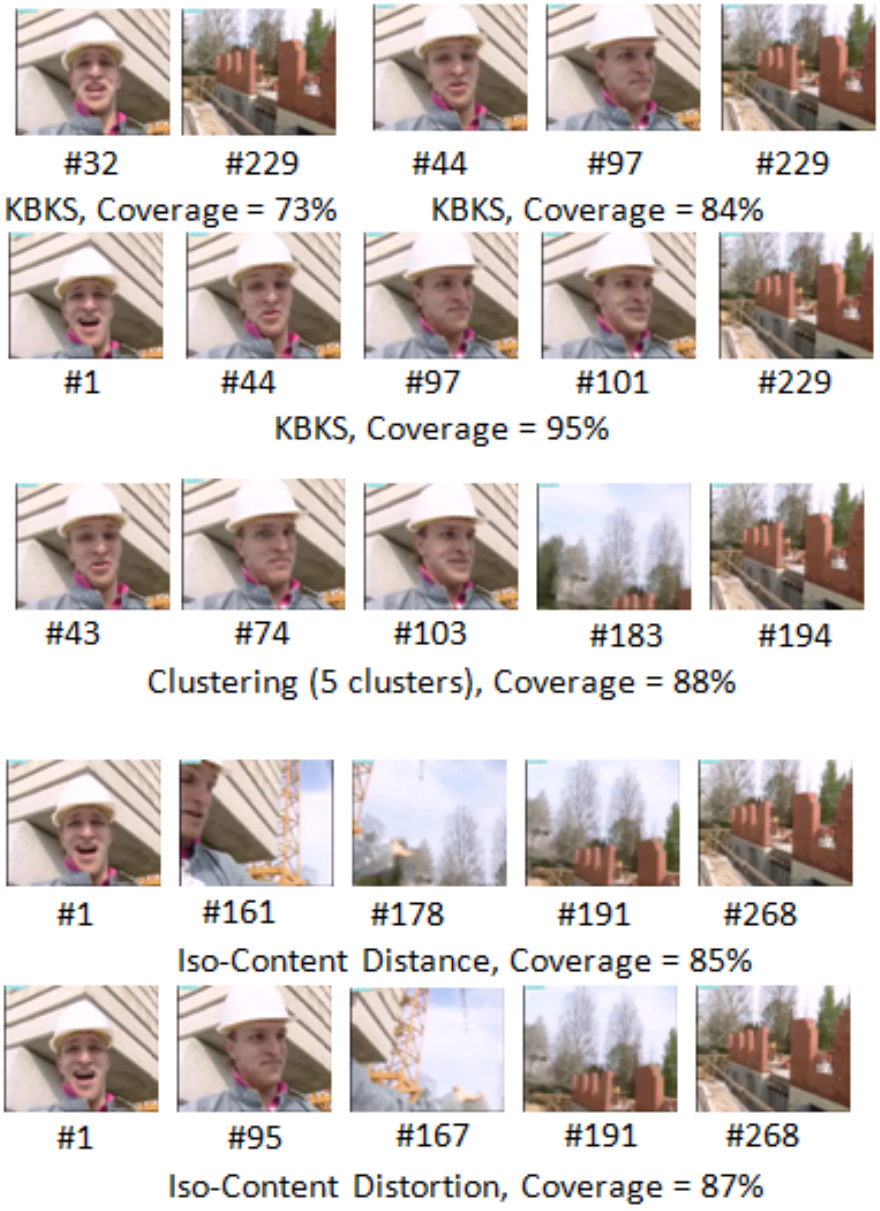


Figure 3.9: Keyframe selection results for the Foreman video.

the end of the shot with a high coverage of 86%, since the frames at the beginning are part of such a keyframe. For the clustering-based method, if the number of cluster is set to 1, we get the keyframe with the middle frame of the shot. That is, clustering based approaches generally take the frame with average information as representative frames. For the Equidistance method, it has the limitation of selecting both the first and the last

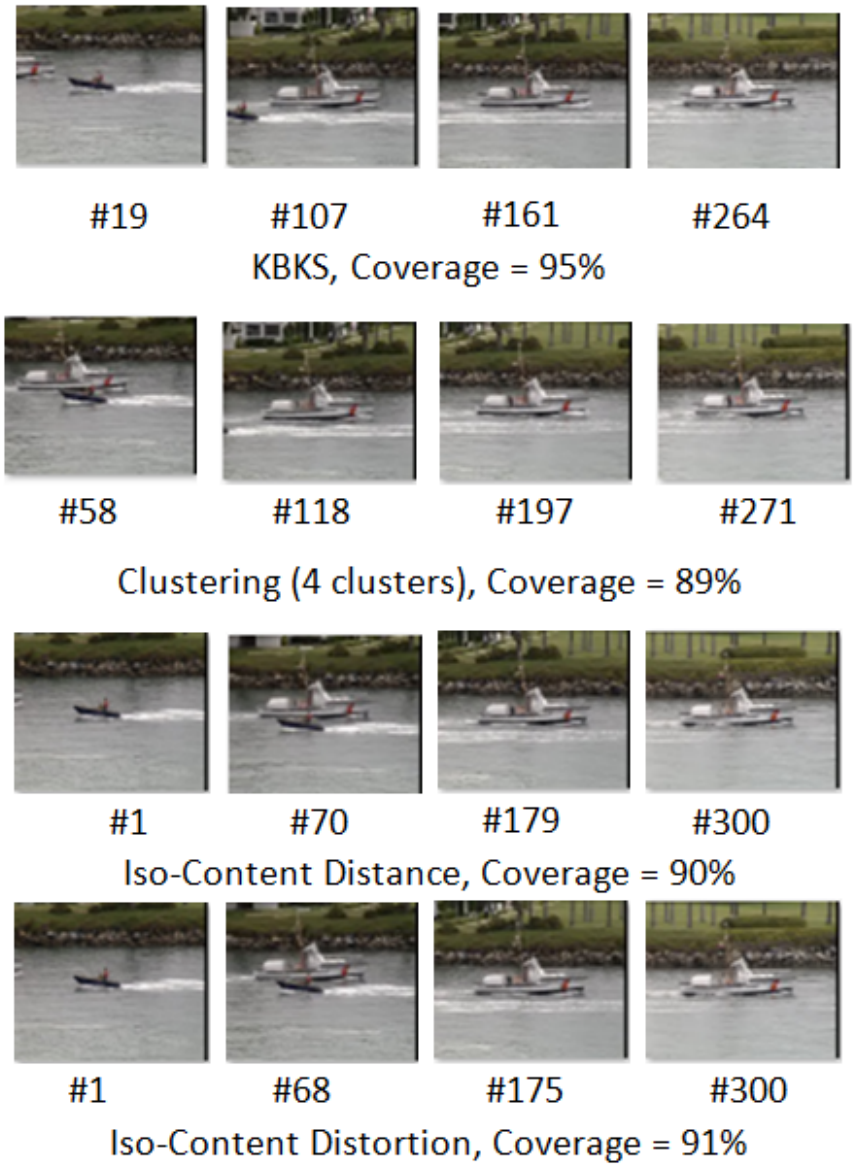


Figure 3.10: Keyframe selection results for the Coastguard video.

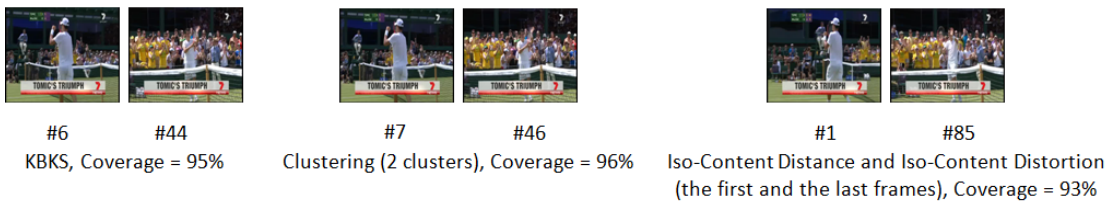


Figure 3.11: Keyframe selection results for the Tennis video.

Figure 3.12: Keyframe selection results for the Zooming video.

frames as a starting point, which is not necessary for many cases such as zooming.

3.4.3 Quantitative Evaluation

The ground-truth keyframes of the videos described in Table 3.1 are manually selected by three students with video processing background. When calculating the metrics, we average the results among the three ground-truth sets of keyframes. The number of target keyframes is set to five. As for our approach, we try different values of coverage starting from 50% until five keyframes are generated. The following metrics are chosen: Precision, Recall, F-score, and Dissimilarity.

A candidate keyframe is considered matched if being no more than X frames apart from a ground truth keyframe. A ground-truth keyframe will be matched with at most one candidate keyframes. F-score is a combination of both the precision and recall indicating the overall quality. Dissimilarity measures the difference between the candidate keyframes and the ground-truth keyframes. It is defined as:

$$Dissimilarity = \sum_{f_c} min_{f_t} d(f_c, f_t), \tag{3.6}$$

where f_c is a candidate keyframe and f_t is a ground-truth keyframe, and $d(f_c, f_t)$ is a distance measure of two keyframes, which is the difference of their frame indices.

In order to explore the influence of X, various experiments were conducted by varying X from 10 to 20 while fixing α to 1 and T to 5. As shown in Fig. 3.13, the F-score of every method increases and stabilizes. While setting a high value for X does not

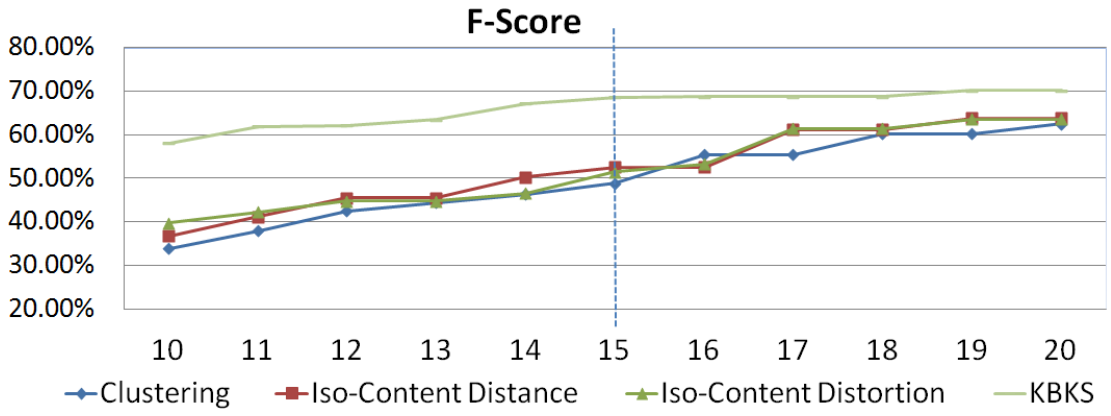


Figure 3.13: Influence of X on the F-score.

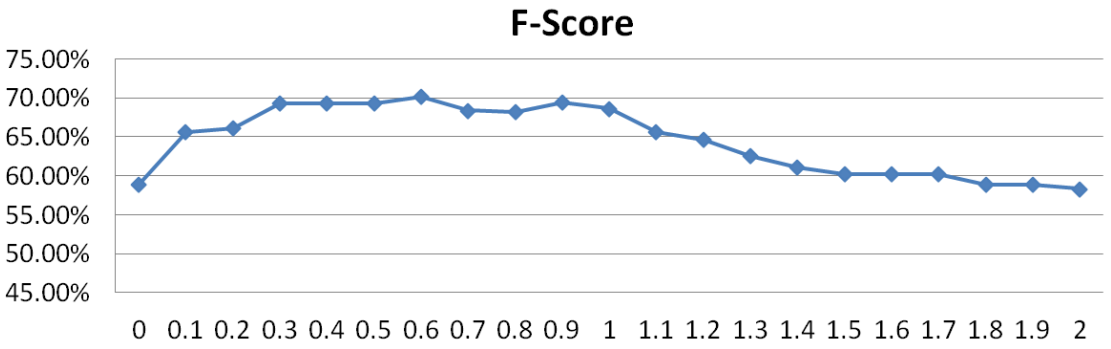


Figure 3.14: Influence of α on the F-score.

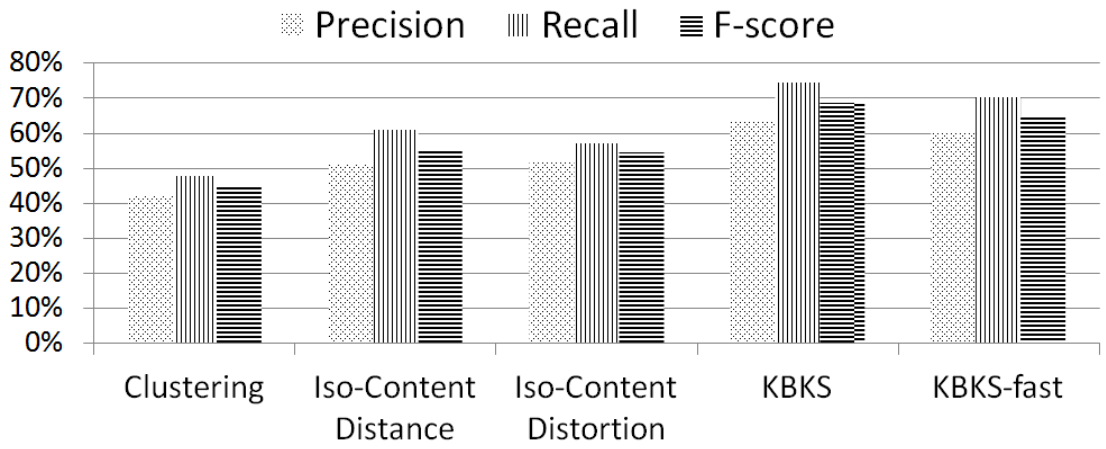


Figure 3.15: Quantitative Evaluation on the second dataset in terms of Precision, Recall and F-score

Table 3.2: Quantitative Evaluation on the second dataset: Dissimilarity

Clustering	Iso-Content Distance	Iso-Content Distortion	KBKS	KBKS-fast
35.3	29.72	30.72	27.5	28.1

reflect a true match, we set X to 15 in the following experiments. Similarly, experiments were conducted to explore the influence of α in (3.3) by setting X to 15 and T to 5, and varying α from 0 to 2. As shown in Fig. 3.14, α does influence the selection result, however, not in a significant way. F-Score grows when α increases from 0 to 0.3, and stabilizes between 0.3 and 1.2. This could be explained that a frame with a higher coverage introduces more new visual content and is more likely to introduce less redundancy. For the sake of simplicity, we set α to 1 in the following experiments.

As illustrated in Fig. 3.15, our approach achieves better performance in regards to precision, recall and F-score. The dissimilarity result shown in Table 3.2 also indicates that the results of our approach are more similar to the ground truth compared to other methods.

3.4.4 Computational Complexity

In our experiment, the frame size of Foreman and Coastguard is 352 x 288, and frame size of the videos in the Open Video project is 352 x 240. With a standard 3.0GHz Dual core desktop computer, for a video shot of 300 frames (i.e., 10 seconds), the total time needed by our algorithm without parallel computing is roughly 150 seconds broken down into 150 seconds for the first step (Section II.A) and the second step (Section II.B) and less than 1 second for the third step (Section II.C) and the fourth step (Section III).

The computational cost of our approach is largely affected by the efficiency of Keypoint Extraction and Matching. As for Keypoint Extraction, it costs about 0.02 second to process one frame. Regarding Keypoint Matching, it takes about 0.1 second to process one frame-pair. Therefore, the time cost of keyframe selection on a video shot with N frame is roughly $N*0.02+W*N*0.1+1$, and complexity is $O(N)$. When $N=300$ and $W=5$, the time cost is about 150 seconds.

In order to reduce the computational cost, we utilized the randomized kd-tree forest based matching algorithm [41] within the window. The matching speed is about ten times faster than the conventional matching algorithm. That is, the computational cost of the fast matching algorithm is about 15 seconds for 300 frames. As shown in the rightmost column of Fig. 3.15 and Table 3.2, the performance of the fast algorithm (namely KBKS-fast) is still comparable to the original scheme, though approximated matching is employed in [41].

3.5 Summary

In this chapter we present a keyframe selection framework based on discriminative keypoints. A video shot is firstly represented by a global pool of keypoints through keypoint

chaining. Secondly, a greedy algorithm is developed to select suitable keyframes based on the two intuitive metrics: Coverage and Redundancy. The experimental results on both case studies and quantitative evaluation demonstrate that our proposed approach is very promising.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

A Top-down Approach for Video Summarization

This chapter proposes an enhanced video summarization framework. It is based on results and includes text that have been published in [42].

4.1 Introduction

While looking into the steps of the experiments in the previous chapter further, we find out that a few problems exist, such as nosiy and unimportant keypoint in the background, the high cost of keypoint matching for every pair of frames (Section 3.2.1), unstable keypoint chain due to inaccurate keypoint matching (Section 3.2.2), and the overall time cost for longer videos (Section 3.4.4). In this chapter, we propose a new top-down approach for video summarization, with steps that address all these issues.

We will start with a brief explanation of Gestalt Psychology, which is a theory of mind of the Berlin School. By taking a holistic stand point, Gestaltism focuses on the

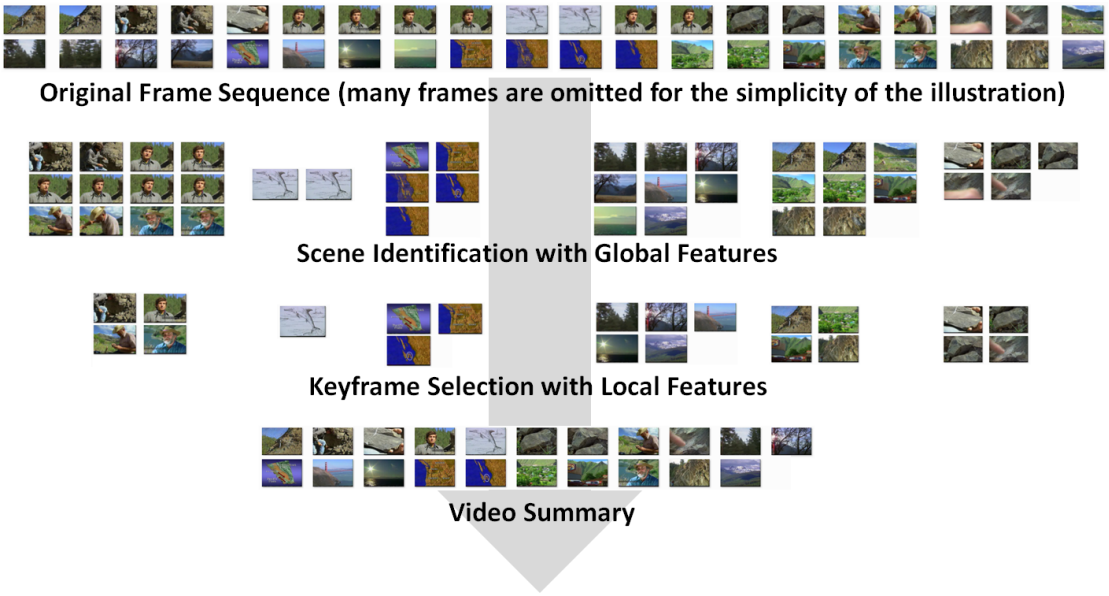


Figure 4.1: Illustration of the proposed top-down approach.

emergent properties of visual stimuli rather than considering them individually. Following its well-known rallying cry, "The whole is greater than the sum of its parts," Gestaltism provides a set of perceptual rules in order to explain that perception cannot be reduced to parts or even to piecewise relations among parts. Such a top-down manner has been neglected in existing video summarization methods so far, since almost all of the previous approaches are designed in a bottom-up fashion to build on the information that is based on the relation between consecutive frames, which are basically the parts of the video, instead of considering the fact that transitions naturally emerge when the video is considered as a whole.

In the current literature, very few of them take both global and local viewpoints into account (see Section 2.1). In general, an edited video depicts a story or an event with a number of scenes (or scenarios, physical environments, fine-grained characteristics of video environmental semantics [43]) in different temporal orders and shooting angles, and video content of the same scene is often visually similar. For unedited raw footage

(e.g., consumer videos), video content will continuously transit from one scene to another, such as panning from the left to right at first and zooming in for a close-up of a person. Meanwhile, human beings often take a top-down approach for storytelling: outlining at first the story or event by identifying the key scenes and then focusing on the details of each scene. Therefore, it would be ideal to take a top-down perspective for video summarization: identifying the scenes of a video at first and performing keyframe selection within each scene next.

In addition, most existing methods share one common feature: each frame is represented with global visual features (e.g., color and texture). Therefore, some subtle yet important details could be shadowed by global features, which results in less representative content in the final video summary. For example, when a person makes faces in a video, traditional video summarization methods may only produce one frame by missing the details of facial dynamics, since global visual features are not able to characterize such fine details. Note that in the context of this work, global features also include the features created by aggregating local features.

Local keypoint features such as scale-invariant feature transform (SIFT) descriptor [4] have played a significant role in many application domains of visual content analysis such as Near-Duplicate Keyframe Detection [44][45], shot boundary detection [46][47] and places clustering in videos [48] due to their distinctive representation capacity (e.g., invariant to location, scale and rotation, and robust to affine transformation). Hence, it would be beneficial to characterize each frame with a number of keypoints and each keypoint is represented with a visual descriptor, and all the frames contribute their keypoints to depict the whole video scene. Therefore, video summarization could be formulated as a problem of identifying a number of frames whose keypoints are representative for the video.

In light of the above observations, in this chapter, we propose a top-down video

summarization framework to account for both the global and local perspectives of a video. As illustrated in Fig. 4.1, video frames are firstly clustered to identify scenes of a video with X-means method [49] and scene summarization is then applied to each scene. The final video summary is an assemble of each scene summary where the selected keyframes are organized in their original temporal order. By assuming that a video scene is depicted with a set of unique keypoints, we formulate scene summarization as a keypoint coverage problem: choosing the keyframes which best cover the unique keypoints and share minimal redundancy. Therefore, we define two criteria: *Coverage* and *Redundancy*, in terms of the keypoints shared among video frames, which is different with other local perspective approaches that explore the difference or importance among adjacent frames. By building the unique keypoint set, a coarsely identified scene can be further discriminated, even it may include multiple semantic scenes. Since it is computationally expensive to build the set of unique keypoints for a scene through keypoint matching and chaining, we further propose a visual word based approach to speed up this process.

Preliminary results of this approach are first reported in [50], then in our later study, we have made three major extensions: 1) for scene identification, we represent each video frame with global visual features and utilize the clustering method X-means [49] to produce scene clusters, instead of K-means, since X-means is computationally scalable, which allows our approach to handle long videos efficiently, and does not require the exact number of final clusters in advance; 2) for scene summarization, we propose a fast algorithm for our previous keyframe selection algorithm reported in [50][33] by developing a kd-tree forest based local visual word model to build the set of unique keypoints for a video; and 3) we conduct more systematic evaluation with two popular video summarization datasets.

In summary, the major contributions in this chapter are as follows.

- Motivated by how video content is captured and organized and how human beings perceive a story or an event, we propose a top-down approach for video summarization: scene identification and scene summarization, which exploits both global features and local details of a video. Note that such a framework allows parallel computing at the scene summarization stage. The main purpose of clustering based scene identification are to improve 1) the efficiency of keyframe extraction by reducing the number of frames to be processed; and 2) the effectiveness of keypoint matching by restricting the matching process within visually similar frames.

- We propose to formulate scene summarization as a keypoint coverage problem by efficiently identifying the set of unique keypoints of a video. To the best of our knowledge, this is one of the first attempts on building a unique keypoint pool for video summarization. By building the unique keypoint set, our method does not critically depend on accurate scene identification. As evidenced with our experiments, our summarization method is able to achieve improved efficiency even with coarsely identified scenes, since keypoints are able to further discriminate different semantic scenes. Such scene summarization method also makes our top-down approach unique, while most existing summarization algorithms generally achieve hierarchical browsing through setting different clustering parameters.

- We propose a kd-tree forest based local visual word model to improve our previous keyframe selection algorithm for scene summarization.

- We conduct comprehensive experiments with two popular video summarization datasets for evaluation and discussions.

The rest of the chapter is organized as follows. In Section 4.2, we explain X-means based scene identification with global visual features. In Section 4.3, we describe the

key steps of the keypoint feature based keyframe selection method to summarize a scene. Experimental results and discussions are presented in Section 4.4, followed by the conclusion in Section 4.5.

4.2 Scene Identification with Global Features

Since the frames of each scene are visually similar, it is straightforward to achieve scene identification by clustering all the frames of a video. Though being very popular due to its simplicity, K-means has three key drawbacks [49]. Firstly, the number of clusters must be pre-determined by users, which is difficult for users when they do not have sufficient prior knowledge of the data. If this number is set inappropriately, the resulting clusters would be either too coarse or too dense. Secondly, it does not scale well computationally for a large amount of data. Thirdly, the clustering result is easily affected by the initialization and it usually converges to local minima which might be far away from the globally best results.

Therefore, X-means algorithm [49] was proposed to address these issues by requesting a range in which the number of clusters K reasonably lies. Basically, it is an iterative process to estimate the best K by starting K-means algorithm with the lower bound of the range. That is, at a stage for a given K value, traditional K-means algorithm is performed on the dataset and a model score is calculated to evaluate the model. As shown in Fig. 4.2, three clusters (i.e., $K = 3$) are produced and the black dots denote centroids of the clusters. At the next stage, all the clusters will be split into two clusters locally by performing 2-means algorithm. The model score will decided whether such a splitting is necessary. As a result, the number of clusters will increase until reaching the upper bound or freeze. Among such an iterative process, the clustering which has the best model score will be selected as the final output. That is, the score model is used in both local centroid splitting and global selection of the final clustering output.

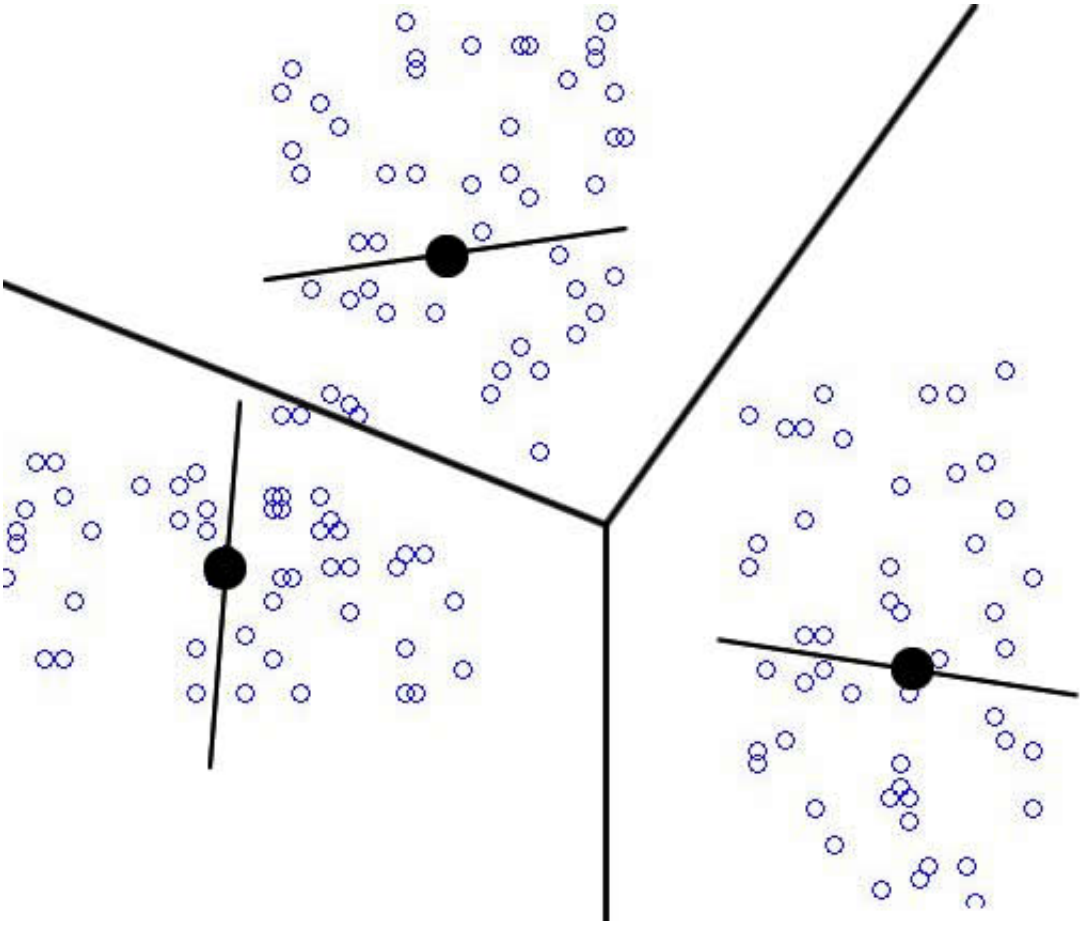


Figure 4.2: Illustration of centroid splitting in X-means clustering algorithm.

The model score is defined with the Bayesian Information Criterion (BIC): .

$$BIC(M_i) = L_i(D) - \frac{p_i}{2} * \log R, \tag{4.1}$$

where M_i is the i-th generated model, D is the input data and R is number of points in D. p_i is the number of parameters in M_i, which is the sum of $K-1$ class probabilities, $M \times K$ centroid coordinates (M is the dimensionality of each point), and one variance estimate. $L_i(D)$ is the log-likelihood of the data according to the i-th model and taken at the maximum-likelihood point. The maximum likelihood estimate for the variance is calculated under the identical spherical Gaussian assumption, which is described in more details in [49]. The second part of the formula is actually a penalty term for the

number of parameters in the model to avoid over-fitting.

As illustrated in the second row of Fig. 4.1, six scenes are identified. We represent each video frame with the Colour and Edge Directivity Descriptor (CEDD) [40], which is a histogram characterizing both color and texture features. Note that other advanced clustering algorithms (e.g., [51][52]), scene identification methods and global features can also be utilized in our framework.

4.3 Scene Summarization with Local Visual Words

Since video frames of a scene are visually similar, keypoint features are important to further characterize and differentiate individual frames. Therefore, a scene can be viewed as a set of unique keypoints and keyframe selection for scene summarization is formulated as below to maximize the coverage of the set of unique keypoints and minimize redundancy among selected frames:

$$\arg\max_{KF}\{\alpha \times Cov(KF) - (1-\alpha) \times Red(KF)\} \tag{4.2}$$

where KF is the final set of selected keyframes in the scene summary, $Cov(KF)$ and $Red(KF)$ are the functions to quantify the coverage of the KF over the set of unique keypoints of the whole scene and redundancy among KF, respectively, and α is a weighting parameter of *Coverage* and *Redundancy*. The details of solving the problem are explained in Section 4.3.4.

As explained in our previous studies [50][33], the key is to construct the set of unique keypoints for a scene and it is very time consuming to achieve this through keypoint matching, tracking and chaining. Therefore, we develop a local visual word based method by constructing a keypoint forest. Noticing that there exist many noisy keypoints which could compromise the performance of keyframe selection, we utilize

saliency map to filter such noisy keypoints and achieve robust performance.

In our work, Lowe's SIFT (Scale-invariant feature transform) descriptor [4] is utilized for keypoint extraction and representation, though many other local features [36] are also applicable for our approach.

4.3.1 Saliency Map based Keypoint Filtering

There are often thousands of local SIFT keypoints within a frame, and not all of these keypoints are visually salient to humans. The second row of Fig. 4.3 shows the locations of keypoints marked as small circles for three sample frames. In order to obtain a robust representation of the video content with local keypoints, we should only keep those with high saliency.

We utilize the saliency detection algorithm proposed in [53] to produce a saliency map (i.e., a saliency value from 0 to 255 for each pixel) for each frame, and filter only those keypoints with saliency lower than a predefined threshold S. As shown in Fig. 4.3, salient areas in the sample frames are automatically detected and highlighted in bright color (shown in the third row), and the prominent objects are reflected in the remaining keypoints after saliency filtering (shown in the fourth row).

4.3.2 Keypoint Forest

After obtaining the filtered keypoints for each frame, a global dictionary from all the keypoints k_x in each frame f_i is to be built to represent the content of the whole scene. In order to find the unique keypoints across all the video frames in the scene, ideally every two frames f_i and f_j should go through keypoint matching using keypoint features. That is, every pair of keypoints between these two frames should be compared to check whether they are identical or not. However, such a straightforward strategy is very computationally expensive. For example, there are about 31,000 (= 250 $\times$ 249 / 2)

Figure 4.3: Illustration of saliency map based keypoint filtering. Original images (1st row), images with SIFT keypoints overlaid (2nd row), saliency maps of original images (3rd row) and images with remaining keypoints overlaid (4th row) of three sample frames.

frame pairs for a 10-second video segment at 25 fps, and each pair of frames requires hundreds of thousands of keypoint comparisons.

To overcome this issue, we adopt the technique proposed in [54] to build a randomized kd-tree forest for all the keypoints in a scene, which enables very fast k-nearest neighbour search in large scale datasets. Essentially, a kd-tree forest is a number of kd-trees that partitions the data in different ways so that the possibility of finding true neighbours are higher than using only a single kd-tree. Partitions are organized into a binary tree with the root element corresponding to the whole space. Each element is then divided into two halves by thresholding along a certain dimension. Both the splitting dimension and the threshold are determined as a statistic of the data records contained in the partition. For a single kd-tree, the splitting dimension is the one having largest sample variance. For a randomized kd-tree forest, each tree adopts different splitting dimensions, e.g., one tree starts with the splitting dimension having the largest variance, and another one starts with the dimension having the second largest variance. The splitting threshold is either the sample mean or the median. Leaves of a tree are atomic partitions and they contain zero or more data records.

4.3.3 Local Visual Word Model

While keypoint forest speeds up k-nearest neighbor search for keypoint matching, local visual word model is developed to accommodate variance of the same keypoint appearing in different frames by grouping neighbouring keypoints into local visual words.

Each keypoint goes through the forest to find the k nearest neighbours. If k is set to a large number, then it may induce many false neighbours; if k is too small, many of the true neighbours may be lost due to the approximation in the algorithm. In the experiment, k is set to 15 empirically to achieve a good balance and the distance between two keypoints is measured with the Euclidean distance between their local descriptors.

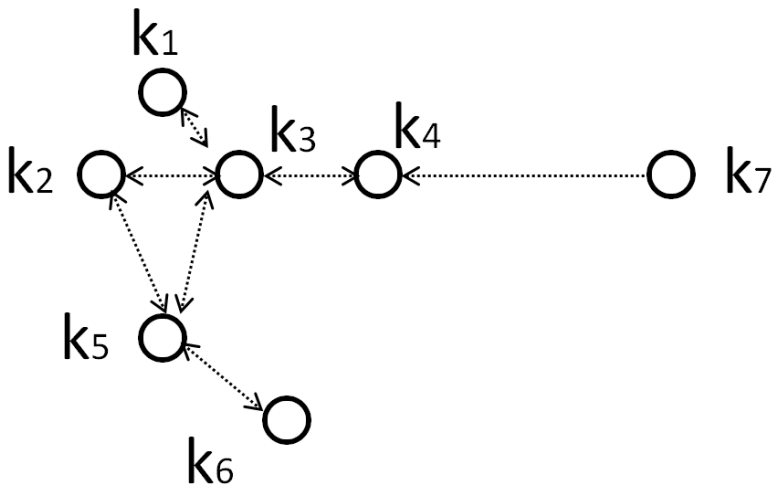


Figure 4.4: Illustration of forming neighbouring keypoint group through mutual neighbourhood among keypoints, where a directional arrow from k_i to k_j means k_i has k_j as its neighbour.

To further reduce false neighbours, the neighbourhood of k_i and k_j is retained only if k_i and k_j are neighbours of each other. If k_i is a noisy keypoint that "mistakenly" has k_j as its own neighbour, it is very likely that k_j's true neighbours have a closer distance to k_j so the neighbourhood between k_i and k_j will not be kept.

Based on the nature of neighbourhood, when a keypoint k_x has a neighbour keypoint k_y, and the same keypoint k_y has a neighbour keypoint k_z, we put all these keypoints into one neighbourhood.

A simplified illustration of forming keypoint neighbourhood is presented in Fig. 4.4. In this toy example, k_1 and k_3 are neighbours of each other, so as k_2 and k_3, k_3 and k_4. As a result, keypoints k_1 to k_6 forms a neighbourhood, while k_7 is excluded from the neighbourhood since k_4 is a neighbour of k_7, but not vice versa.

After forming the keypoint neighbourhoods, each keypoint either belongs to a neighbourhood or becomes an singleton with no neighbours. All singleton keypoints, which are very likely to be noise keypoints, are removed. Each neighbourhood N_i corresponds to a group of keypoints with high visual similarity and is denoted as a Local Visual

Word LVW_{N_i} for all the keypoints in N_i. Because the number of keypoints in a neighbourhood reflects its importance in a scene, we set a threshold T on this number to further filter less important/unstable local visual words.

Compared to the traditional visual word generation where vector-based visual features are clustered into a predefined number of visual word, which might significantly lose the discriminative power of local keypoints, local visual word model emphasizes the quality of the keypoint matching by forming small and growing keypoint neighbourhoods, so that a representative local visual word dictionary can be created based on the strong similarity within a neighbourhood.

4.3.4 Keyframe Selection

The goal of keyframe selection is to best represent a scene with a minimal number of frames. That is, the keyframes should be able to best represent the visual content of a scene while minimizing redundant content among them. In our case, to ensure the best representation, the keypoints of those keyframes should cover the global dictionary of local visual words $LVW_{dictionary}$ as much as possible. Since it is a NP-complete problem, we approach it in a greedy manner.

First of all, the global dictionary is separated into two sets, $LVW_{covered}$ and $LVW_{uncovered}$. At the beginning of the process, $LVW_{uncovered}$ contains all local visual words in $LVW_{dictionary}$ and $LVW_{covered}$ is empty. Since each keypoint in a frame is mapped to either a local visual word during the formation of keypoint neighbourhood, or nothing, each frame f_i has a local visual word set denoted as lvw_i. If multiple keypoints in a frame correspond to the same local visual word, only one of them is counted to form a set of unique keypoints, though such counting information can be further utilized to derive the importance of each keypoint.

The *Coverage* $C(f_i)$ of the frame f_i to the dictionary can then be defined as the sum

of the weights of the intersected local visual words between lvw_i and the uncovered set $LVW_{uncovered}$.

$$lvwC_i = lvw_i \cap LVW_{uncovered}, \tag{4.3}$$

$$C(f_i) = \sum_{lvw \in lvwC_i} |lvw|, \tag{4.4}$$

where $lvwC_i$ is the intersection of a frame's local visual word set and the global uncovered set, and $|lvw|$ is the number of keypoints in a local visual word lvw.

Likewise, *Redundancy* $R(f_i)$ of the frame f_i is defined as the sum of the weights of the intersected local visual words between lvw_i and the covered set $LVW_{covered}$, reflecting how redundant it is based on the already-covered content in the scene:

$$lvwR_i = lvw_i \cap LVW_{covered}, \tag{4.5}$$

$$R(f_i) = \sum_{lvw \in lvwR_i} |lvw|. \tag{4.6}$$

The influence of frame f_i at an iteration is calculated in Equation (4.7) as a linear combination of $C(f_i)$ and $R(f_i)$ controlled by α.

$$Influence(f_i) = \frac{\alpha * C(f_i) - (1-\alpha)R(f_i)}{GlobalSim(f_i)}, \tag{4.7}$$

where

$$GlobalSim(f_i) = \sum_j Similarity(f_i, f_j). \tag{4.8}$$

That is, the influence of a frame f_i will be increased if it shares low similarity (i.e., small $GlobalSim(f_i)$) with other frames in terms of its global visual feature. This factor is added because some frames with a large portion of uniform regions (e.g., sky) do not

have a reasonable number of keypoints.

At each iteration, the frame with the highest influence value will be selected as a keyframe, and $LVW_{covered}$ and $LVW_{uncovered}$ will be updated accordingly based on the selected keyframe:

$$LVW_{covered} = LVW_{covered} \cup lvw_i, \tag{4.9}$$

$$LVW_{uncovered} = LVW_{uncovered} - lvw_i. \tag{4.10}$$

The iteration repeats until a predefined percentage of coverage $STOP$ over the $LVW_{dictionary}$ is reached.

4.4 Experiments and Discussions

4.4.1 Experimental Settings

We perform experiments on two datasets. The first dataset is the one used in [9], [10] and [11], which contains 50 videos from the Open Video Project (OVP)[1], and user ground-truth summaries provided by [9]. Each video is about 100 seconds long and has 3036 frames on average. The second dataset is provided by [9][2], which contains 50 videos covering several genres (cartoons, news, sports, commercials, tv-shows and home videos). Their duration varies from 1 to 10 minutes. More detailed statistics of this dataset can be found at [9].

Our proposed approach (namely KFVSUM) is compared with Sparse Dictionary based approach (SD) [18], VSUMM [9], OVP storyboard, Delaunay Clustering approach (DT) [11] and STIMO [10]. We implement the SD approach, since experimental

[1] http://www.open-video.org

[2] https://sites.google.com/site/vsummsite/download

results are available for VSUMM, OVP, DT, and STIMO approaches in the first dataset, and available for VSUMM only in the second dataset. The top 10 frames selected by Sparse Dictionary based method [18] is used for evaluation to comply with its experimental settings, since the average number of ground truth keyframes is around 10. And the tuning parameter λ of SD approach is set to 0.15 to achieve the best summarization result.

In our experiments, each video is down-sampled at 5 frames per second (fps), which is comparable to the practice of other studies (e.g., [12] at 5 fps, [11] at 3 fps and VSUMM [9] at 1 fps). Because there are much redundant information in neighbouring frames, so down-sampling can be applied to reduce computational cost without losing important information. The number of trees Trs in the kd-tree forest is set to 5 to balance accuracy and efficiency for finding keypoint neighbours.

4.4.2 Evaluation Metrics

Automatic summaries (AS) generated by different algorithms are compared with all the user summaries (US) to obtain quantitative assessment. Three evaluation metrics, precision, recall, and F-score, are used to measure summarization quality of each algorithm:

$$Precision = \frac{n_{matched}}{n_{AS}}, \tag{4.11}$$

$$Recall = \frac{n_{matched}}{n_{US}}, \tag{4.12}$$

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{4.13}$$

where $n_{matched}$ is the number of matching keyframes (discussed below) from an automatic summary, n_{AS} is the number keyframes in the automatic summary and n_{US} is the

number of keyframes in a user summary.

In [9], two frames are matched if the Manhattan distance of their color histograms is less than a predetermined threshold δ. In this chapter, we use a stronger matching criteria than [9]. Combined with requirement on the visual similarity, we also impose a temporal requirement that two frames are considered matched only if they are no more than Δ frames apart from each other. We set $\delta = 0.5$ and $\Delta = 60$ (i.e., 2 seconds) in our experiments. Moreover, a user summary keyframe will be matched with at most one candidate keyframe in an automatic summary, so multiple keyframes matching the same user summary keyframe will be penalized.

4.4.3 Impact of Parameter Settings

We randomly withhold half videos (i.e., 25 videos) of the first dataset as a validation dataset to investigate the impact of different parameters for our proposed approach and to guide our comparison study.

The upper bound parameter of X-means, K, determines the maximal number of resultant clusters. The saliency threshold S is used to remove background keypoints in a frame. The threshold T to filter the local visual word affects the size of the dictionary and thus the granularity of details it captures. α is used to balance the contribution of *Coverage* and *Redundancy* in Eqn. 4.7. $STOP$ is the measurement of summarization quality in terms of coverage percentage. Analytical tests are performed to identify appropriate values for these parameters.

As shown in Fig. 4.5, F-score value varies little while K is greater than 10. Therefore, we set the lower bound of the range for the X-means algorithm to 2, and the upper bound to 10 throughout our experiments.

It is noticed that the method proposed in [53] does a good job detecting salient objects in a frame, so object segmentation can be achieved with satisfactory results.

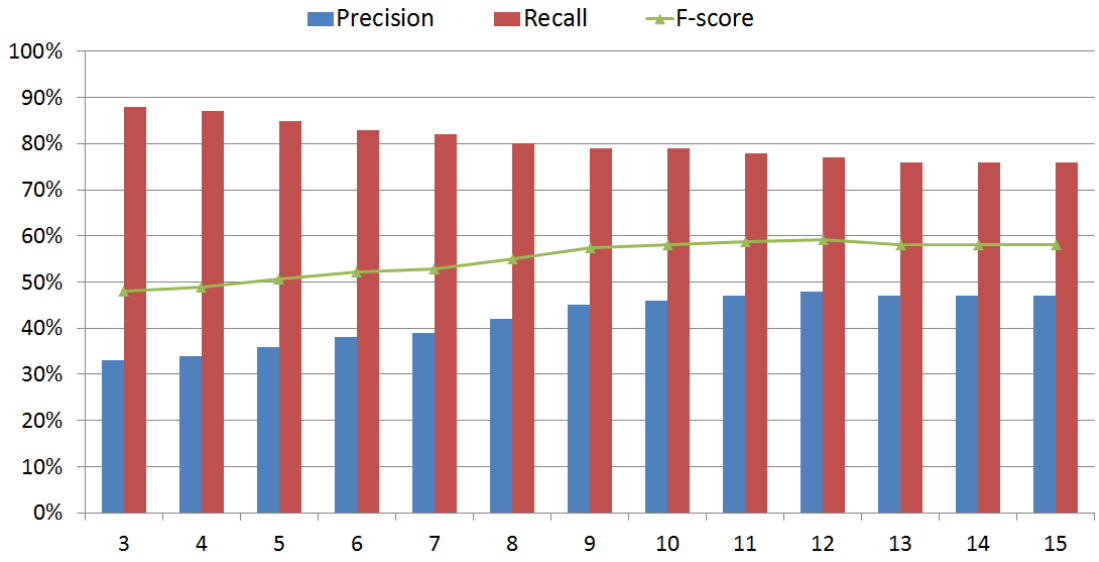


Figure 4.5: Impact of K, the upper bound of X-means algorithm.

However, in many cases it also assigns high saliency values to areas with strong texture such as rocks and buildings. As shown in Fig. 4.6, experiments were performed when $S = 0, 50, 100, 150, 200, 250$ while fixing T to 20 and α to 0.5. F-score grows slightly with increasing S, but drops significantly when S reaches a very high value (i.e., 250), where most of the keypoints are discarded so few meaningful local visual words are generated. It also shows that removing low-saliency keypoints has some positive impact on the performance. In the following experiments, we set S to 200 to keep the most keypoints in the foreground, though some background keypoints may still exist after saliency filtering.

In order to explore the impact of T, experiments were conducted by varying T from 1 to 40 while fixing S to 200 and α to 0.5. As shown in Fig. 4.7, when T is small, most of the local visual words including noisy visual words are kept, which results in a dictionary with a large size and requires a large number of frames to best cover the dictionary. As a result, the precision is very low and recall is very high. On the other hand, when T is large, most of the local visual words are discarded, and very few frames will be selected. Based on this observation, T is set to 20 in other experiments. That is,

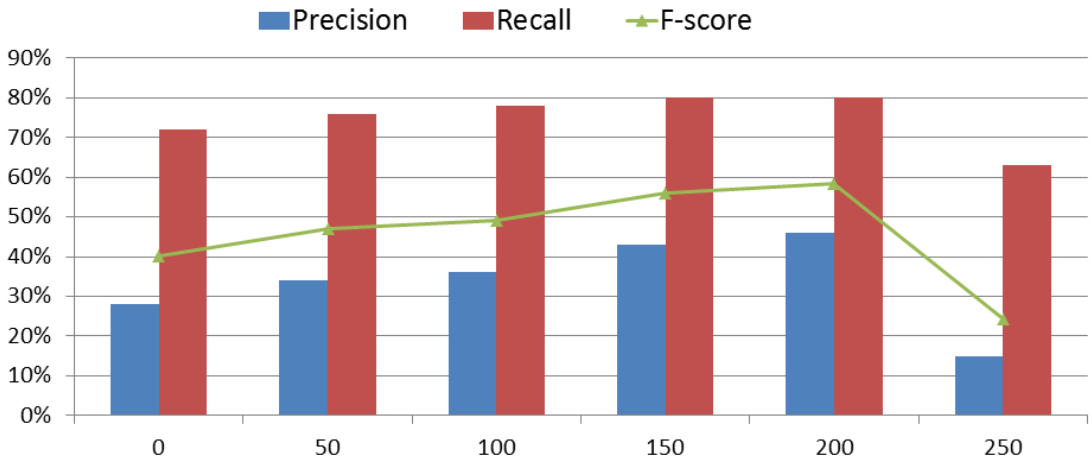


Figure 4.6: Impact of S, the saliency threshold.

keypoint neighbourhood with fewer than 20 keypoints are discarded.

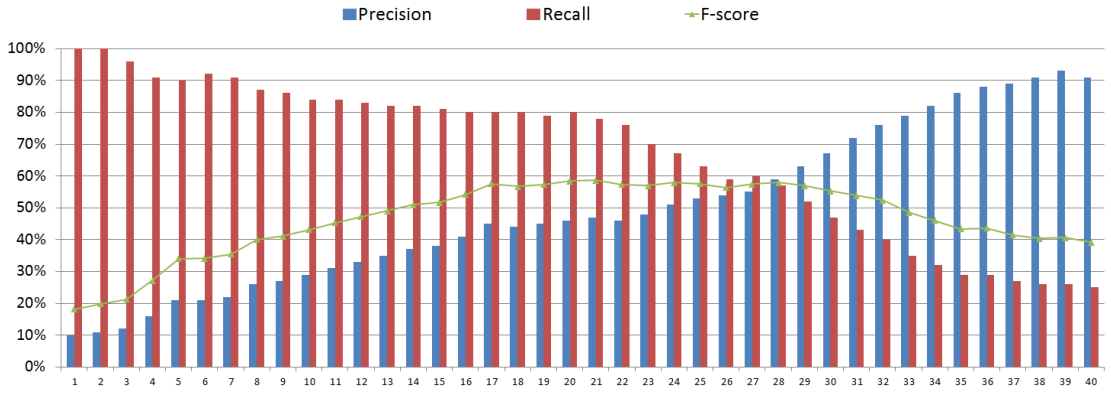


Figure 4.7: Impact of T, local visual word neighbourhood threshold.

As shown in Fig. 4.8, F-score grows when α increases from 0.1 to 0.5, so it indicates that *Coverage* is an important factor in the selection. On the other hand, F-score stabilizes at higher value of α. This could be explained that a frame with a higher *Coverage* introduces more new visual content and is more likely to introduce less *Redundancy*. For the sake of simplicity, we set α to 0.5.

As shown in Fig. 4.9, precision decreases clearly, when $STOP$ is set to greater than 85%, which also compromises F-score, though recall increases. This is because that greater $STOP$ values will increase the number of keyframes selected, which decreases

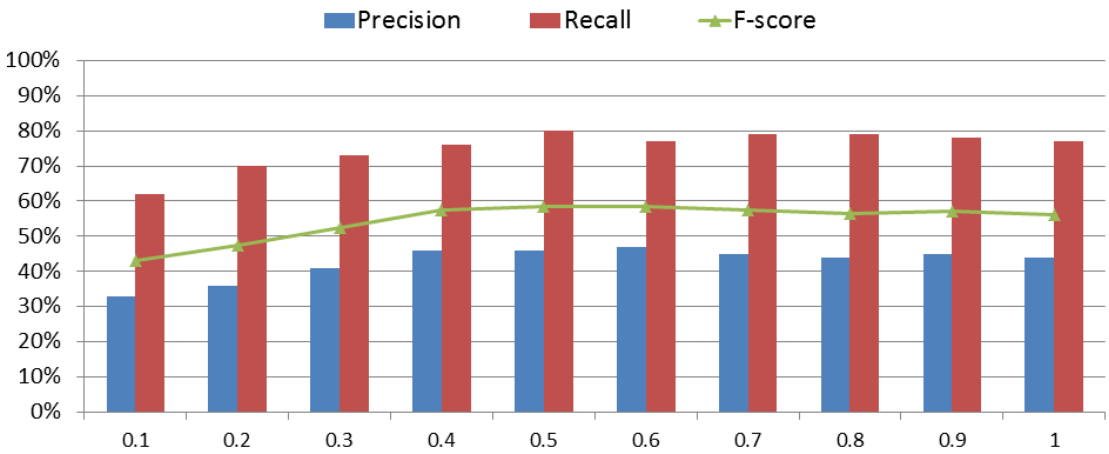


Figure 4.8: Impact of α, the weight between coverage and redundancy.

Table 4.1: Performance of Each Method for the First Dataset.

	Precision	Recall	F-score
OVP	43%	64%	51%
DT	47%	50%	49%
STIMO	39%	65%	49%
VSUMM	42%	77%	54%
SD	40%	61%	48%
KPVSUM	46%	63%	53%
KFVSUM	46%	79%	58%

precision and increases recall. Therefore, we set the dictionary coverage threshold $STOP$ to 85% for the experiments of comparison study.

4.4.4 Performance Evaluation

For the first dataset, as illustrated in Fig. 4.10 and detailed in Table 4.1, our approach achieves the best performance among all compared methods in terms of F-score. By

Table 4.2: The Average Number of Selected Keyframes.

OVP	DT	STIMO	VSUMM	SD	KPVSUM	KFVSUM
9.66	6.2	9.96	9.92	10	9.16	9.76

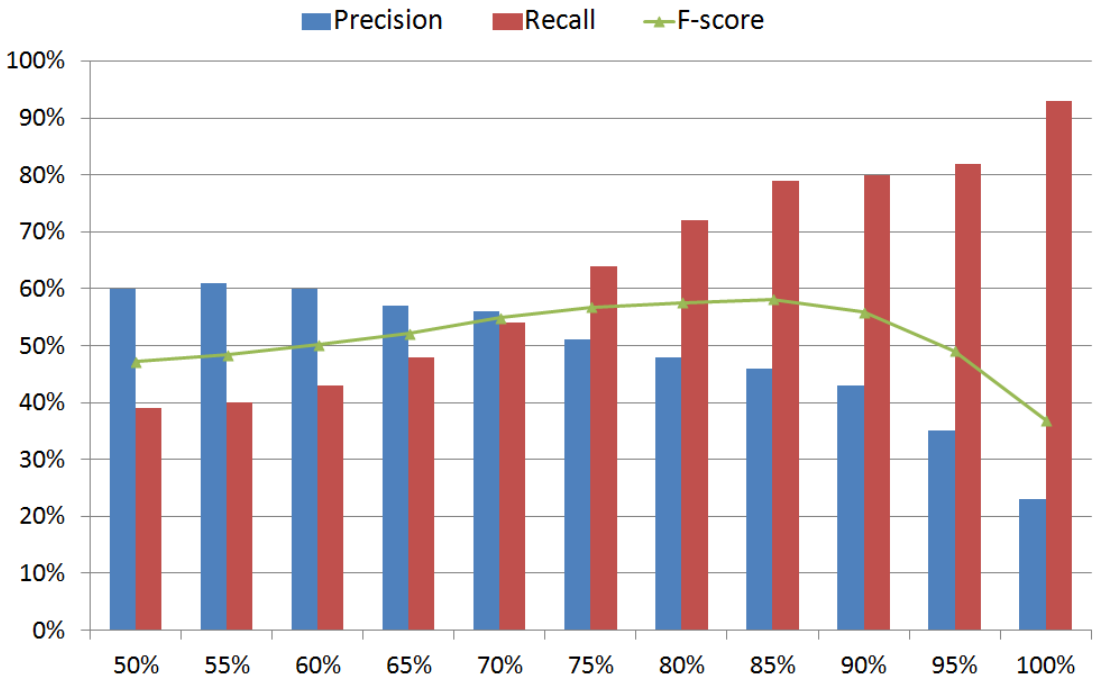


Figure 4.9: Impact of *STOP*, the coverage percentage of the final summary.

further analysing the selected keyframes in terms of the average number of selected keyframes (shown in Table 4.2), it is noticed that DT selects fewer keyframes than others, so they have slightly higher precision values and lower recall rates due to the limited selection of keyframes, while OVP, STIMO, VSUMM, SD, KPVSUM [50], and our approach KFVSUM produce similar numbers of keyframes on average. Similarly, for the second dataset, our proposed approach also outperforms all other methods in terms of F-score, as illustrated in Fig. 4.11 and detailed in Table 4.3.

The effectiveness of scene identification is shown at Fig. 4.12 for the first dataset. Without scene identification, keypoint kd-trees are generated for the whole video, so the size of trees would become bigger, and it may cause more false keypoint neighbours. Thus a bigger local visual word dictionary would be generated with possibly more noisy local visual words, and eventually compromise the performance of keyframe selection. In addition, a global dictionary for the whole video might cause selection bias towards the scenes with more visual details, so short scenes and scenes with less visual content

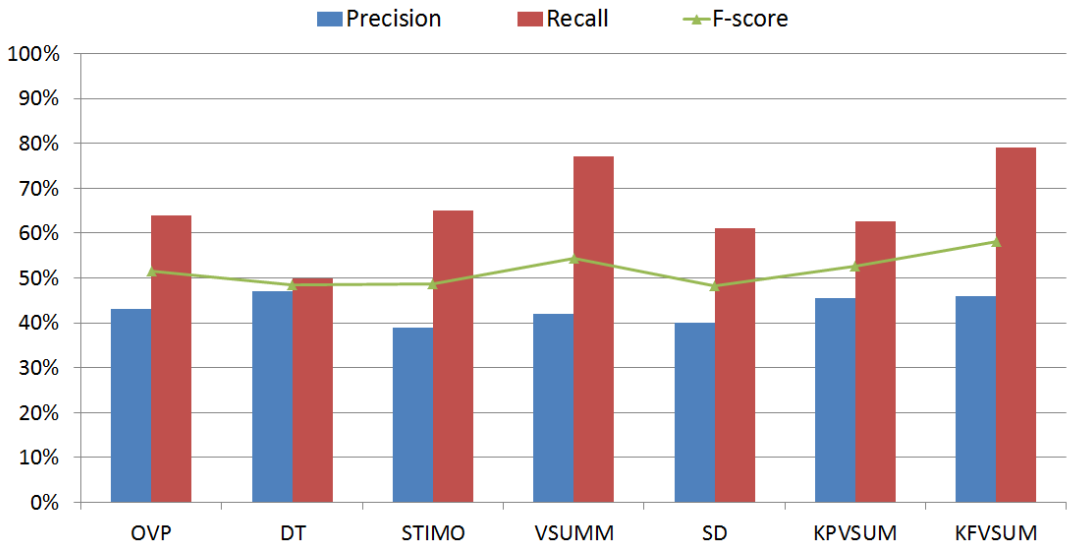


Figure 4.10: Quantitative evaluation in terms of Precision, Recall and F-score for the first dataset.

Table 4.3: Performance of Each Method for the Second Dataset.

	Precision	Recall	F-score
VSUMM	38%	72%	50%
SD	37%	53%	44%
KFVSUM	42%	74%	54%

would be neglected. As a result, the performance degrades without grouping the frames into scenes, as more keyframes with similar visual contents are produced and some true keyframes are missing.

In our experiments, the frame size of the videos in the Open Video project is 352 x 240, and the frame size of the videos in the Youtube dataset is 320 x 240. With a standard 3.0GHz Dual core desktop computer, for a video shot of 60 seconds (300 frames to be processed with a sampling rate of 5fps), the total time needed by our algorithm without parallel computing is roughly 10 seconds, broken down into 1 seconds for the scene identification with global features (Section 4.2), 6 seconds for local keypoint extraction [4] which requires roughly 0.02 second for each frame, 2 seconds for saliency map based keypoint filtering (section 4.3.1) which requires roughly 0.007 second for

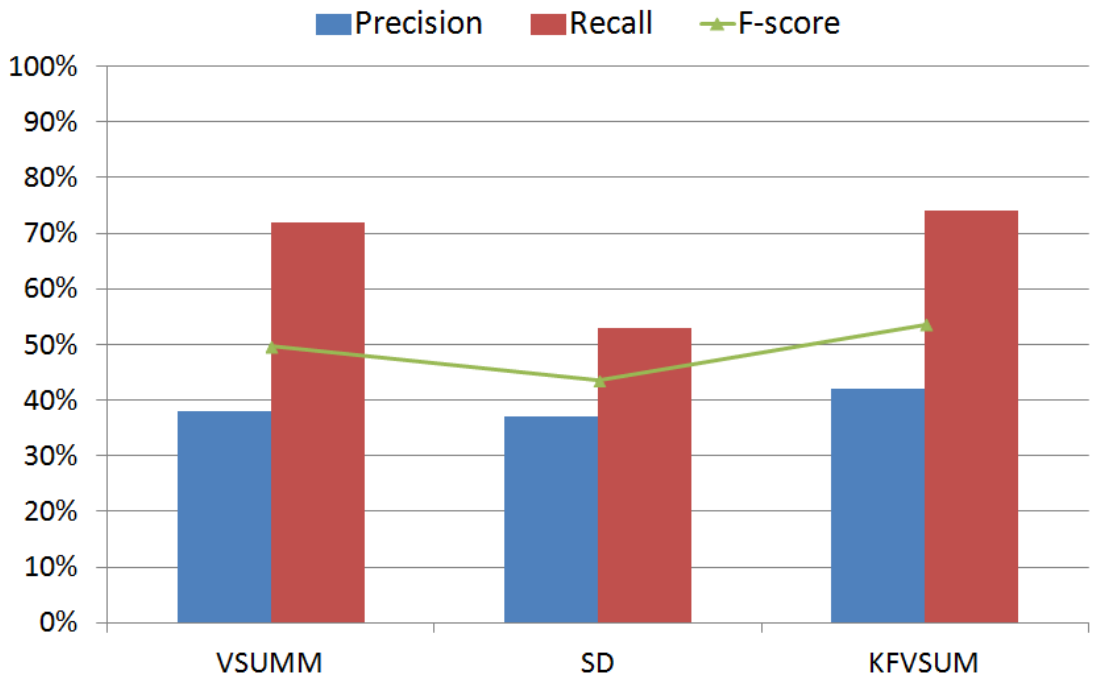


Figure 4.11: Quantitative evaluation in terms of Precision, Recall and F-score for the second dataset.

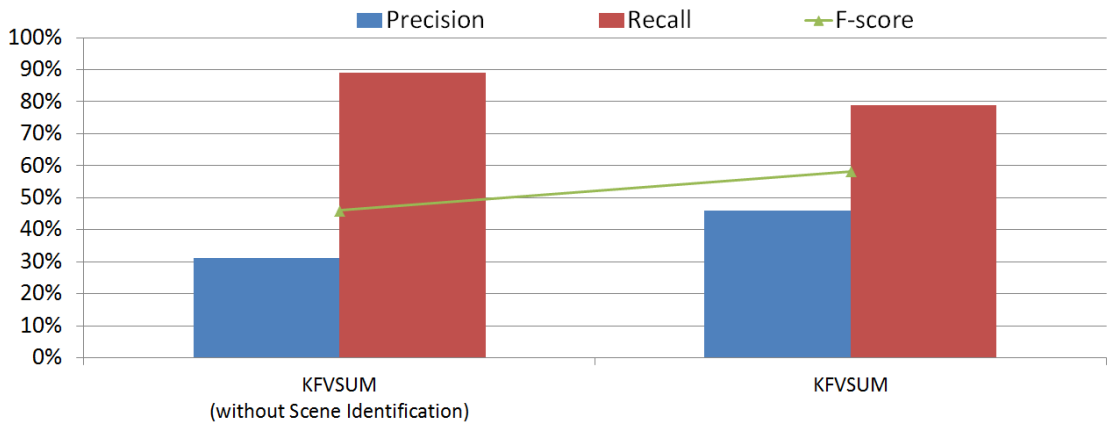


Figure 4.12: Effectiveness of scene identification.

each frame, 1 second for building the keypoint forest and performing keyframe selection. This speed is about 15 times faster than our previous approach [33]. For SD based approach [18], the most computationally expensive part is to obtain sparse dictionary and sparse representation, which takes about 1.5 minutes for a 60-second video.

4.4.5 Case Studies

This section presents three summarization samples for subjective evaluation: the first two samples from the first dataset and the third one from the second dataset. The first example is from the 4th video of the first dataset. The video summarization results of all the methods are shown in Fig 4.13. As illustrated, DT tends to choose fewer keyframes than others. In addition, since DT is a clustering-based method, it normally selects keyframes representing average information within a cluster (i.e., close to the centroid of a cluster), which might cause problems when a cluster contains frames from multiple scenes. In that case, it will likely select keyframes during a fading transition that are most similar to all the frames in the cluster, as clearly shown in Fig 4.13. This problem of DT is also observed from many other videos in our experiments. Being a human selected summary, OVP is able to roughly cover the content, but there are still some details missing. Comparing KFVSUM with STIMO, VSUMM, and SD, we notice that there are missing keyframes from STIMO, VSUMM and SD. In particular, the last two keyframes being selected by KFVSUM contain more details than those in VSUMM, such as the tree in the second last keyframe and the text titles in the last keyframe. This indicates the effectiveness of keypoint based scene summarization algorithm.

The second example is from the 15th video of the first dataset. The video summarization results of all the methods are shown at Fig 4.14. OVP contains two very similar frames at the second and the second last positions, because keyframes are manually

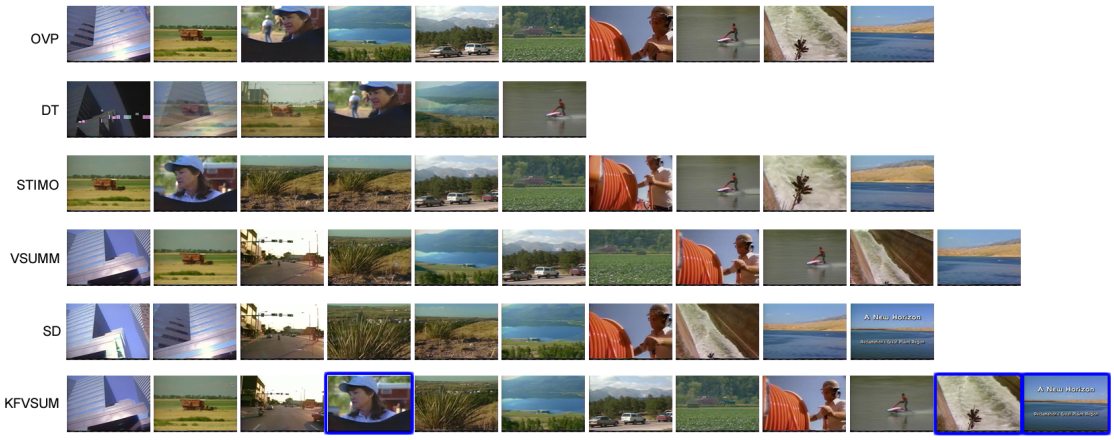


Figure 4.13: Sample video summarization results of all the methods for the 4th video of the first dataset, from top to bottom: OVP, DT, STIMO, VSUMM, SD, and KFVSUM (ours).

selected by humans based on the temporal order of frames. Therefore, content redundancy is introduced in the final summary, and DT, STIMO, VSUMM and SD misses a few keyframes. In addition, DT and STIMO select the keyframes with limited local details, such as the keyframes framed in black and orange, because global visual feature are only able to discriminate significant difference between those pure-background frames and other frames. Since our method utilizes the local details in a frame, non-informative frames are discarded. On the other hand, the two frame pairs highlighted in blue and red rectangles, though having similar global visual features, are extracted in our method but missed in all the others. It is also noticed that our approach misses the last frame identified by SD, which could be explained that its local visual words as well as global features are similar to those of other frames with textual content.

The third sample is from the second video (v12) of the second dataset. As shown in Fig. 4.15, the first three keyframes of all the four methods are very similar. Note that VSUMM has two redundant keyframes, while missing some important frames. Though performing better than VSUMM, SD misses two important frames (e.g., the 6th and 7th frames in the KFVSUM row).

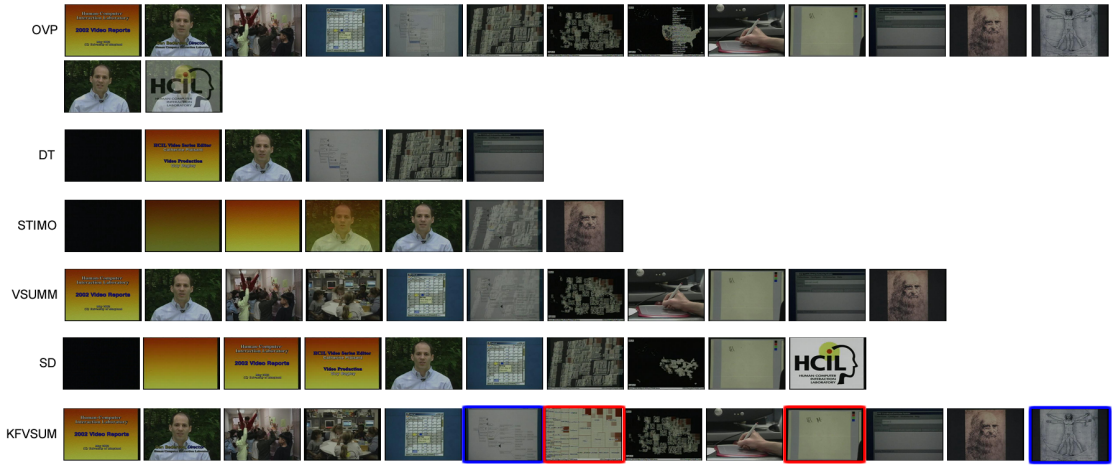


Figure 4.14: Sample video summarization results of all the methods for the 15th video of the first dataset, from top to bottom: OVP, DT, STIMO, VSUMM, SD, and KFVSUM (ours).

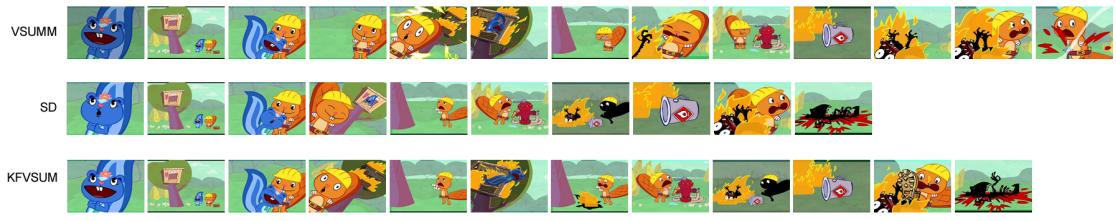


Figure 4.15: Sample video summarization results for the 2nd video of the second dataset, from top to bottom: VSUMM, SD, and KFVSUM (ours).

4.4.6 Summaries with Different Lengths

Having a local visual word dictionary for each scene, we can easily customize our desired summary. Within each scene of a video, the number of selected keyframes is not predefined, but is determined by the desired coverage of contents. A static/low-motion scene may produce fewer keyframes with very high coverage of the scene. On the other hand, scenes with substantial change of contents may require more keyframes to reach the same level of coverage. Therefore, we could adjust the length of the summary by tuning the desired coverage of content $STOP$ denoted by percentage, which is more intuitive than setting a fixed number, and is more adaptive to different kinds of videos. This flexibility is mostly lacked in the previous works.

Fig. 4.16 to Fig. 4.19 demonstrate the flexibility of our approach by adapting to summaries with different coverages and lengths. As summaries with different lengths can be prepared, hierarchical skimming can also be supported as a top-down overview of a video. In addition, a different coverage threshold can be given to different scenes, so that it is possible to further customize the granularity of the summary for each scene automatically or via user interaction.

Figure 4.16: Video summarization with $STOP = 60\%$.

Figure 4.17: Video summarization with $STOP = 70\%$.

Figure 4.18: Video summarization with $STOP = 80\%$.

4.5 Summary

In this chapter we present a top-down approach video summarization framework by exploiting both visual similarity among the frames within a scene to identify scenes within a video and local details for scene summarization. Video frames firstly are automatically grouped into scenes with global visual features and representative frames of each scene are identified with local features. In addition, we formulate scene summarization

Figure 4.19: Video summarization with $STOP = 90\%$.

as a coverage problem in terms of keypoints of a scene. Rather than performing time-consuming keypoint matching, tracking and chaining, a keypoint forest based approach is proposed to construct a dictionary of local visual words for efficient scene summarization. Our proposed two metrics *Coverage* and *Redundancy* are intuitive for users to fine tune the final summary. The experimental results demonstrate that our proposed approach outperforms the state-of-the-art and is flexible to generate various lengths of keyframe based summaries. In the future, we will discover the importance of each scene and individual local descriptors and incorporate such information into our framework. It would also be interesting to explore spatial and temporal context for more efficient keypoint matching and to investigate the impact of various local features.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Video Summarization via Minimum Sparse Reconstruction

This chapter proposes another video summarization approach based on minimum sparse reconstruction (MSR) principle. It is based on results and includes text that have been published in [55].

5.1 Introduction

Video summarization has been extensively studied and there exist a large number of methods in the literature as detailed in Chapter 2. Recently, Cong *et al.* [56] formulated video summarization from a new point of view. In this algorithm, video summarization is transformed into a sparse dictionary (SD) selection problem and an $L_{2,1}$ norm based relaxing constraint is imposed to ensure sparsity. Since the $L_{2,1}$ norm based constraint cannot ensure sparsity directly, the SD cannot be selected directly as keyframes for video summarization. Instead, they are selected by identifying local maximums of an importance curve generated according to the norm of reconstruction coefficients. As a result, the selected keyframes are not the optimal subset of frames for the model

since the frames with least reconstruction coefficients, which also contribute to reducing reconstruction errors, may not be selected. In addition, the optimization algorithm for dictionary selection is computational expensive, which makes it not suitable for real-time applications.

In this chapter, with the inspiration from the SD based algorithm [56], video summarization is formulated as a problem selecting the minimum number of keyframes to reconstruct the entire video as accurate as possible. And the real sparse constraint L_0 norm, instead of the relaxing constraint $L_{2,1}$ norm, is adopted to ensure sparsity. A selection matrix is proposed to model the selection of keyframes from the original video, according to which the L_0 norm of this selection matrix is proposed to ensure selecting as few keyframes as possible. As a result, SD can be selected directly as keyframes for video summarization. Specifically, two computationally effective video summarization algorithms based on minimum sparse reconstruction (MSR) principle, including an off-line version and an on-line version, are proposed to extract keyframes for video summarization. In addition, a percentage of reconstruction (POR) criterion is also proposed to summarize video sequence with different length, enabling the proposed MSR based video summarization algorithms adaptive to different kinds of videos. Finally, experiments on two benchmark datasets are conducted to demonstrate the effectiveness of the proposed algorithms.

The main contributions of this chapter reside in three aspects:

1. An MSR based video summarization model is formulated by utilizing a selection matrix, such that video summarization is performed by utilizing minimum number of keyframes to reconstruct the entire video as accurate as possible. An L_0 norm based constraint is imposed to ensure real sparsity such that keyframes are selected directly according to the selection matrix.

2. Two efficient and effective MSR based video summarization algorithms are proposed for off-line and on-line applications, respectively.

3. A scalable strategy is designed to provide flexibility for practical applications. The proposed POR criterion can be tuned to extract a keyframe set at different levels of reconstruction of the original video sequence.

5.2 Problem Formulation

5.2.1 Minimum Sparse Reconstruction Constrained Video Summarization Model

The key element for our video summarization is to select an optimal subset from the entire video frame pool through which the original video can be reconstructed as accurate as possible. Given an initial candidate pool $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n] \in I\!R^{d \times n}$, where each column vector $\mathbf{f}_i \in I\!R^d$ denotes the feature vector of the i-th video frame. Our goal is to find an optimal subset $\mathbf{F}_K = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{n_k}] \in I\!R^{d \times n_k}$ such that the original frame set $\mathbf{F}$ can be accurately reconstructed by $\mathbf{F}_K$ and the size of $\mathbf{F}_K$ is as small as possible. Therefore, the following minimum sparse reconstruction (MSR) model can be constructed:

$$\begin{aligned} & \min_{\mathbf{S}} : \frac{1}{2}\|\mathbf{F} - \mathbf{F}_K\mathbf{A}\|_2 + \lambda \cdot \|\mathbf{S}\|_0 \\ s.t. \quad & \mathbf{F}_K = \mathbf{F}\mathbf{S} \\ & \mathbf{A} = f(\mathbf{F}, \mathbf{F}_K), \end{aligned} \tag{5.1}$$

where $\mathbf{S}$ is a sparse diagonal matrix defined as *selection matrix*. Its diagonal elements are either '1' or '0', indicating that the corresponding frames are selected as keyframes or not. Therefore, $\|\mathbf{S}\|_0$ represents the number of elements selected from $\mathbf{F}$ for $\mathbf{F}_K$. $\mathbf{A}$ represents the reconstruction coefficients of $\mathbf{F}$ by $\mathbf{F}_K$ using the reconstruction function

$f(\mathbf{F}, \mathbf{F}_K)$. And $\|\cdot\|_2$ represents the L_2 norm of a matrix or vector. The first part in the optimization function of (5.1) is to decrease the least-square reconstruction error (LSRE) as much as possible, while the second part to confine the number of keyframes (n_K) as small as possible. Therefore, the proposed model ensures that the whole video can be reconstructed as accurate as possible with as few keyframes as possible, which is denoted as MSR constrained video summarization model.

5.2.2 Video Frame Representation

In the proposed MSR based video summarization model defined by (5.1), each frame is represented by its feature vector, and thus a candidate pool is formed by compiling the features of all frames. In this chapter, the 360-dimensional feature vector utilized in the SD algorithm [56] is adopted for better performance evaluation, though many other features can also be employed in our method. This feature set consists of two parts: a 252-dimensional feature vector extracted by CENTRIST [57][58], and a 108-dimensional feature vector accounting for color moment.

The CENTRIST feature captures local structures of an image without color information by utilizing a spatial pyramid structure. Only the last two spatial levels, each of which contains 5 and 1 image patches, are adopted. Thus, each patch is represented by a 42-dimensional feature, where 40 dimensions are for eigenvectors and the other two are for the mean and variance of each patch, respectively. Hence, the dimension of each CENTRIST feature is $6 \times 42 = 252$. More details can be found in [57][58].

To calculate the color moment feature of a video frame, each frame is represented in HSV color space and partitioned in to 3×4 patches. A three-order color moment (i.e., Mean, Standard Deviation, and Skewness) is adopted. As a result, each frame is represented with a $3 \times 4 \times 3 \times 3 = 108$-dimensional color moment.

5.3 Solution and Algorithm Implementation

5.3.1 Minimum Sparse Reconstruction Algorithm for Video Summarization

As observed in the video summarization model defined by (5.1), there is a competition between minimizing LSRE and n_K. If we want to decrease the LSRE, the n_K will increase. For example, when n_K equals to the size of original frame candidate pool, the LSRE will be 0, indicating that no video summarization is conducted. Therefore, a balance (denoted by λ) should be achieved between LSRE and n_K. However, it is difficult to select a suitable λ for the optimization problem defined by (5.1), since the magnitudes of LSRE and n_K are not the same and will vary for different videos. Therefore, in this chapter, we take an iterative approach to solve the MSR problem.

Assume that m keyframes $\mathbf{F}_K = [\mathbf{f}_{k_1}, \mathbf{f}_{k_2}, \ldots, \mathbf{f}_{k_m}] \in I\!R^{d\times m}$ have been identified, where $k_1, k_2, \ldots, k_m \in \{1, 2, \ldots, n\}$. Then choosing the next keyframe should be able to maximally decrease the reconstruction error defined in equation (5.1). As a result, the candidate of the next keyframe will be the one producing the maximum reconstruction error at the current iteration:

$$\mathbf{f}_{k_{m+1}} = \arg \max_{\mathbf{f}_j \in \mathbf{F}/\mathbf{F}_K} \|\mathbf{f}_j - \mathbf{F}_K \mathbf{a}_j\|_2, \tag{5.2}$$

in which $\mathbf{a}_j (j = 1, 2, \ldots, n)$ represents the reconstruction coefficients of the j-th frame, and $\mathbf{F}/\mathbf{F}_K$ represents all the non-keyframes.

Due to the fact that feature vectors with high magnitudes are more likely to result in large reconstruction errors, we define the percentage of reconstruction (POR) as the

objective criterion for keyframe selection:

$$\mathbf{f}_{k_{m+1}} = \arg\min_{\mathbf{f}_j \in \mathbf{F}/\mathbf{F}_K} POR_j = \arg\min_{\mathbf{f}_j \in \mathbf{F}/\mathbf{F}_K} \frac{\|\mathbf{f}_j - \mathbf{F}_K \mathbf{a}_j\|_2}{\|\mathbf{f}_j\|_2}. \tag{5.3}$$

Now for each frame in the candidate pool, reconstruction coefficients should be obtained for calculating its POR. Though many methods can be employed for such a reconstruction problem, we use one of the most effective methods, the Orthogonal Subspace Projection (OSP). In OSP, all the frames are projected to the orthogonal space spanned by $\mathbf{F}_K$, and the reconstruction coefficients of the j-th frame are determined as follows:

$$\mathbf{a}_j = (\mathbf{F}_K^T \mathbf{F}_K)^{-1} \mathbf{F}_K^T \mathbf{f}_j = \mathbf{P}_K \mathbf{f}_j, \tag{5.4}$$

in which $\mathbf{P}_K = (\mathbf{F}_K^T \mathbf{F}_K)^{-1} \mathbf{F}_K^T$ is the orthogonal projector of $\mathbf{F}_K$. As a result, the reconstruction error (RE) of the j-th frame is determined by the L_2 norm of orthogonal complement projection of $\mathbf{F}_K$:

$$RE_j = \|\mathbf{f}_j - \mathbf{F}_K \mathbf{a}_j\|_2 = \|(\mathbf{I} - \mathbf{F}_K (\mathbf{F}_K^T \mathbf{F}_K)^{-1} \mathbf{F}_K^T) \mathbf{f}_j\|_2 = \|\mathbf{P}_K^+ \mathbf{f}_j\|_2, \tag{5.5}$$

in which $\mathbf{P}_K^+ = \mathbf{I} - \mathbf{F}_K (\mathbf{F}_K^T \mathbf{F}_K)^{-1} \mathbf{F}_K^T$ represents orthogonal complement projector of $\mathbf{F}_K$ and $\mathbf{I}$ is the unit matrix. Thus, the POR of the j-th frame is defined as follows:

$$POR_j = \frac{RE_j}{\|\mathbf{f}_j\|_2}. \tag{5.6}$$

As a result, the original iterative video summarization model (defined in equation(5.3)) can be rewritten as:

$$\mathbf{f}_{k_{m+1}} = \arg\min_{\mathbf{f}_j \in \mathbf{F}/\mathbf{F}_K} POR_j. \tag{5.7}$$

Note that other constraints can also be further imposed to ensure the reconstruction more physically meaningful, such as sum-to-one constraint which confines the sum of

all the reconstruction coefficients of a frame to one, and nonnegative constraint which requires all the reconstruction coefficients to be nonnegative.

The next task is to set the termination criterion for such an iterative process. The straightforward solution is to predefine the number of keyframes. However, it is still an open issue to determine a reasonable value for the number of keyframes without prior knowledge [33]. In this chapter, we rely on a threshold for POR, i.e. T_{POR}. When the POR of all the frames in a video exceeds T_{POR}, the iteration will terminate and all the extracted keyframes form the final keyframe set.

When deriving the iterative algorithm, we assume that a set of keyframes have been identified. Therefore, in order to provide a complete solution, the initialization issue should be addressed. If prior knowledge exists, we can incorporate it into our algorithm. For example, the cover image of a video can be chosen as the initial keyframe. Otherwise, it would be ideal to select the most representative frame as the first keyframe. In this chapter, we employ the following two strategies to address the initialization issue: 1) selecting the frame with the largest magnitude since it will be more likely to result in large RE:

$$\mathbf{f}_{k_1} = \arg\max_{\mathbf{f}_j \in \mathbf{F}} \|\mathbf{f}_j\|_2, \tag{5.8}$$

and 2) selecting the frame closest to the average of all the frames since it will be more likely to reduce RE of all other frames:

$$\mathbf{f}_{k_1} = \arg\min_{\mathbf{f}_j \in \mathbf{F}} \|\mathbf{f}_j - \bar{\mathbf{f}}\|_2, \tag{5.9}$$

in which $\bar{\mathbf{f}} = \frac{1}{n}\sum_j \mathbf{f}_j$.

5.3.2 Off-line MSR based Video Summarization

We use the word off-line to describe the scenario that summarization will be performed on a given video. That is, all the video frames are available when summarization algorithm starts. Actually, most of the existing video summarization methods focus on such an off-line scenario. By referring to the solution introduced in the above section, off-line MSR based video summarization method (OffMSR) is summarized in Algorithm 1:

Algorithm 1: The Off-line MSR based video summarization algorithm

Input: the whole video frame set $\mathbf{F} \in I\!R^{d\times n}$ and POR threshold T_{POR}.
Output: the keyframe set of the summary output $\mathbf{F}_K \in I\!R^{d\times p}$.
Initialization:
1) Selecting initial keyframes, such as the first keyframe $\mathbf{f}_{k_1}$ according to (5.8) or (5.9).
2) Calculating the POR of all the frames according to (5.7).
3) Setting the iteration counter m to 1.
Iteration for video summarization:
While the POR of any frame is less than T_{POR}:
1) Determine the next keyframe $\mathbf{f}_{k_{m+1}}$ according to (5.7).
2) Update keyframe set $\mathbf{F}_K = [\mathbf{F}_K, \mathbf{f}_{k_{m+1}}]$.
3) Increase iteration counter $m = m+1$.
4) Calculate RE of all the frames by the current keyframe set according to (5.5).
5) Calculate POR of all the frames by the current keyframe set according to (5.6).

A summarization example with an airplane video from the OVP is illustrated in Fig. 5.1. The first keyframe is selected according to the magnitude of all the frames. Then the worst reconstructed frame (the frame with the least POR) is selected as a new keyframe at each iteration until the POR of all the frames exceeds a predefined threshold (70% in this example).

The proposed OffMSR algorithm has the following merits:

- The computational time efficiency of our proposed OffMSR algorithm is more

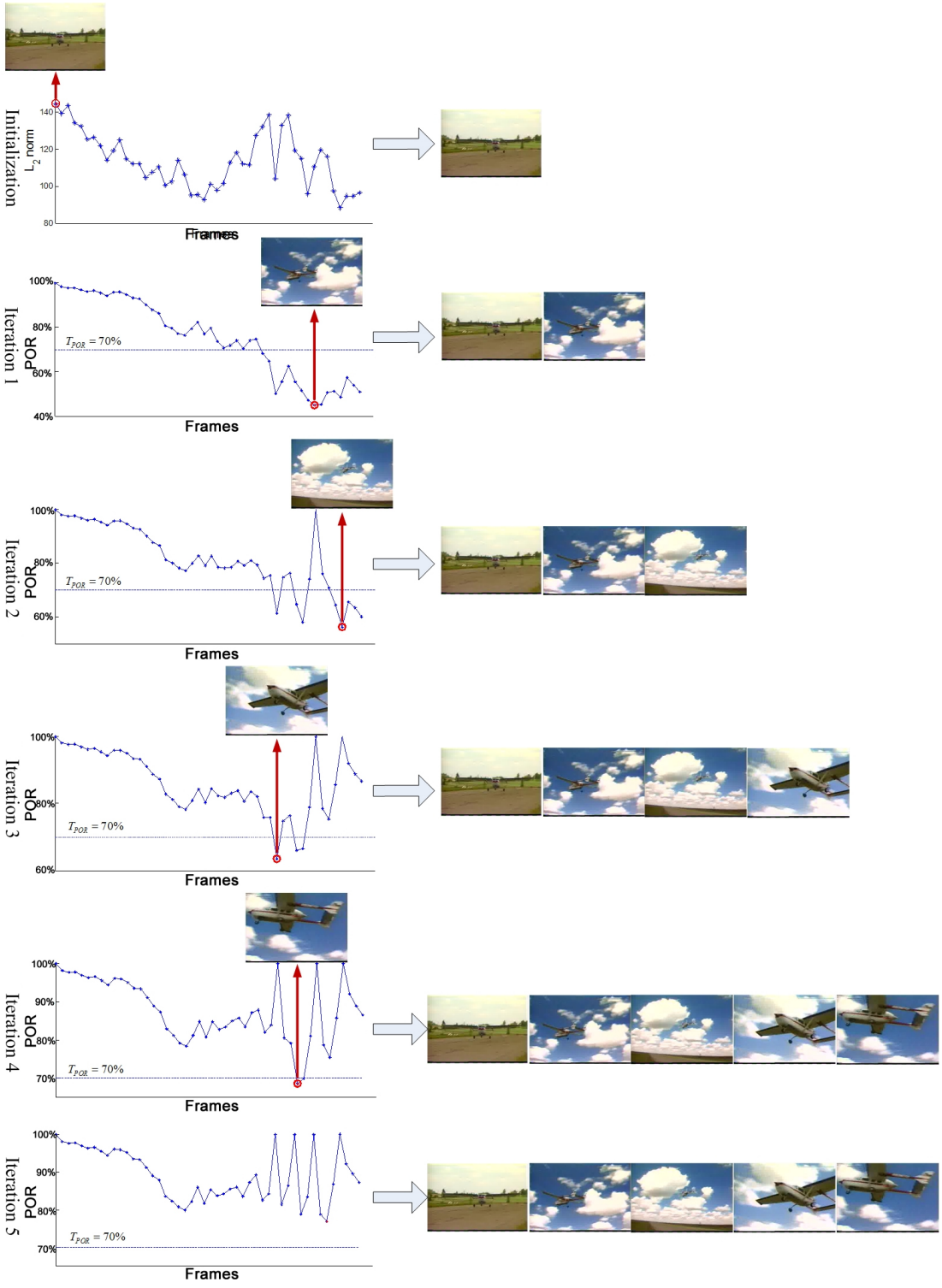


Figure 5.1: A sample illustration of our proposed Off-line MSR based video summarization algorithm. In this example, the frame with maximum magnitude is selected as the first keyframe and $T_{POR} = 70\%$.

Table 5.1: Computation complexity analysis of the $(m+1)$-th $(m=1,2,\ldots)$ iteration in the proposed OffMSR algorithm.

Calculation		Complexity
RE_j	$\mathbf{F}_m^T\mathbf{F}_m$	$O\left(m^2d\right)$
	$(\mathbf{F}_m^T\mathbf{F}_m)^{-1}$	$O\left(m^3\right)$
	$\mathbf{F}_m(\mathbf{F}_m^T\mathbf{F}_m)^{-1}$	$O\left(m^2d\right)$
	$\mathbf{F}_m(\mathbf{F}_m^T\mathbf{F}_m)^{-1}\mathbf{F}_m^T$	$O\left(d^2m\right)$
	$\mathbf{I}-\mathbf{F}_m(\mathbf{F}_m^T\mathbf{F}_m)^{-1}\mathbf{F}_m^T$	$O\left(d^2\right)$
	$(\mathbf{I}-\mathbf{F}_m(\mathbf{F}_m^T\mathbf{F}_m)^{-1}\mathbf{F}_m^T)\mathbf{f}_j$	$O\left(d^2\right)$
	$\|(\mathbf{I}-\mathbf{F}_m(\mathbf{F}_m^T\mathbf{F}_m)^{-1}\mathbf{F}_m^T)\mathbf{f}_j\|_2$	$O(d)$
	Total	$\max\{O\left(m^2d\right),O\left(d^2m\right)\}$
POR_j		$\max\{O\left(m^2d\right),O\left(d^2m\right)\}$
$POR_j(j=1,2,\ldots,n)$		$\max\{O\left(m^2dn\right),O\left(d^2mn\right)\}$
$\mathbf{f}_{k_{m+1}}$		$\max\{O\left(m^2dn\right),O\left(d^2mn\right)\}$

efficient than other known alternative approaches. As observed from equation (5.7), the denominator for calculating POR_j is the L_2 norm of the feature vector and does not change in the iteration. Therefore, it can be stored as constant after the first keyframe is identified. Consequently, in each iteration, only the reconstruction errors of all the frames (defined by $RE_j(j=1,2,\ldots,n)$ in (5.5)) are calculated. Then a minimum calculation is performed for the reconstruction error weighted by the reciprocal of the L_2 norm of the feature vector. As observed from Table 5.1, the computation complexity of the $(m+1)$-th iteration in the proposed OffMSR algorithm is $\max\{O\left(m^2dn\right),O\left(d^2mn\right)\}$. Generally, to extract n_K keyframes, only n_K-1 iteration is required. Therefore, the total computation complexity of the proposed OffMSR algorithm is $\max\{O\left((n_K-1)^2dn\right),O\left(d^2(n_K-1)n\right)\}$. Therefore, the proposed OffMSR algorithm is suitable for real-time applications.

- Compared with the SD based algorithm in [56], our proposed OffMSR algorithm extracts a sparse dictionary as keyframes directly by ensuring sparsity explicitly.

- Our proposed OffMSR algorithm is flexible to produce summaries with different

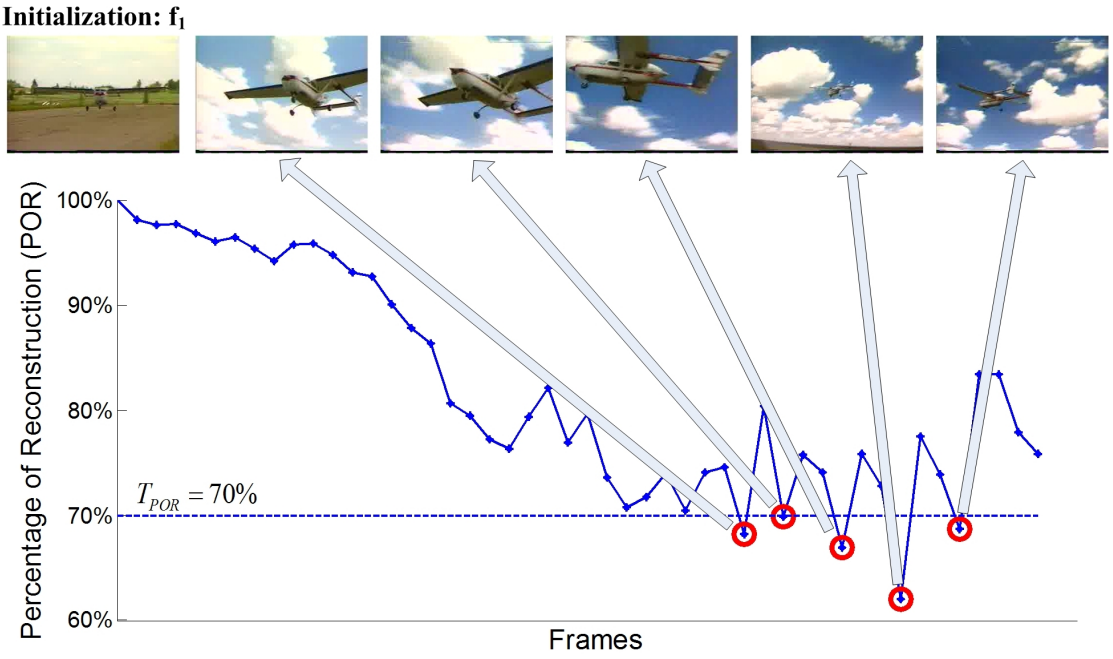


Figure 5.2: A sample illustration of our on-line MSR based video summarization ($T_{POR} = 70\%$).

lengths by intuitively tuning the parameter T_{POR}.

5.3.3 On-line MSR based Video Summarization

On-line MSR based video summarization algorithm is for continuous incoming video streams such as surveillance videos where the input frame set is continuously changing. As shown in Fig. 5.2, for the continuous video stream, the first frame $\mathbf{f}_1$ is first selected as the keyframe $\mathbf{f}_{k_1}$, and thus the current keyframe set $\mathbf{F}_K = [\mathbf{f}_1]$. When a new frame $\mathbf{f}_j (j = 2,3,\ldots)$ is available, it is projected to the subspace spanned by the existing keyframes:

$$RE_j = \|(\mathbf{I} - \mathbf{F}_K(\mathbf{F}_K^T\mathbf{F}_K)^{-1}\mathbf{F}_K^T)\mathbf{f}_j\|_2. \tag{5.10}$$

The POR of the j-th frame is determined as follows:

$$POR_j = \frac{RE_j}{\|\mathbf{f}_j\|}, j = 2,3,\ldots \tag{5.11}$$

If its POR is less than a predefined threshold T_{POR}, the current frame will be selected as a new keyframe: $\mathbf{F}_K = [\mathbf{F}_K, \mathbf{f}_j]$; Otherwise, the current frame $\mathbf{f}_j$ will not be selected and the keyframe set $\mathbf{F}_K$ does not change. Such an on-line strategy can be formulated as follows:

$$POR_j \begin{cases} < T_POR & \text{new keyframe:} \quad \mathbf{F}_K = [\mathbf{F}_K, \mathbf{f}_j] \\ \geq T & \mathbf{F}_K \text{ remains unchanged} \end{cases} \tag{5.12}$$

The proposed on-line MSR based video summarization algorithm (denoted by On-MSR) for real-time video summarization is summarized in Algorithm 2.

Algorithm 2: The On-line MSR based video summarization algorithm

Input: A video sequence $\mathbf{f}_j (j = 1,2,\ldots)$, and POR threshold T_{POR}.
Output: the keyframe set of the summary output $\mathbf{F}_K \in I\!R^{d\times p}$.
Initialization:
Selecting $\mathbf{f}_1$ as the first keyframe $\mathbf{f}_{i_1}$.
Iteration for video summarization:
For $j = 2,3,\ldots$:
1) Calculate the RE of the current frame according to (5.10).
2) Calculate the POR of the current frame represented by current keyframes according to (5.11).
3) Determine whether the current frame is selected as a keyframe according to (5.12).

According the computational complexity analysis shown in Table 5.1, in the proposed OnMSR algorithm, the computational complexity to determine whether the j-th ($j = 1,2,\ldots$) frame is selected as a keyframe or not is $\max\{O\left(m^2 d\right), O\left(d^2 m\right)\}$, in which m is the number of current keyframes in $\mathbf{F}_K$. Obviously, the computation cost is mainly affected by the number of current keyframes m. Generally, m is much smaller

than the number of available video frames n. Even for the continuous video acquiring applications (e.g., video surveillance) that m may continue to increase. Note that keeping m small by removing some early obtained keyframes periodically will help to maintain reasonable computation complexity, which makes the proposed OnMSR algorithm very suitable for real-time applications.

Since the OnMSR algorithm relies on the initial keyframe set, selecting an appropriate set of initial keyframes is very important. In order to reduce the bias of selecting the first frame as the initial keyframe, we can utilize the proposed OffMSR algorithm shown in Algorithm 1 to produce a keyframe set from a video buffer and use the keyframe set as the initial keyframes:

$$\mathbf{F}_{init} = \text{OffMSR}\left(\mathbf{F}_{buff}\right), \tag{5.13}$$

in which $\mathbf{F}_{init}$ represents the initial keyframes extracted from a video buffer, $\mathbf{F}_{buff}$ represents the video buffer consisting of first several frames of a video.

5.4 Experiments and Discussions

In this section, we present various experiments and comparisons to validate the effectiveness and efficiency of our proposed MSR based video summarization algorithm. The experimental settings and evaluation metrics have been introduced in Section 4.4.1 and Section 4.4.2 respectively.

5.4.1 Performance Evaluation

The first dataset with human-selected ground-truth keyframes from five users is available at the VSUMM [59] official website. When calculating quantitative metrics, the average of the results among the five ground-truth sets of keyframes is adopted. Our

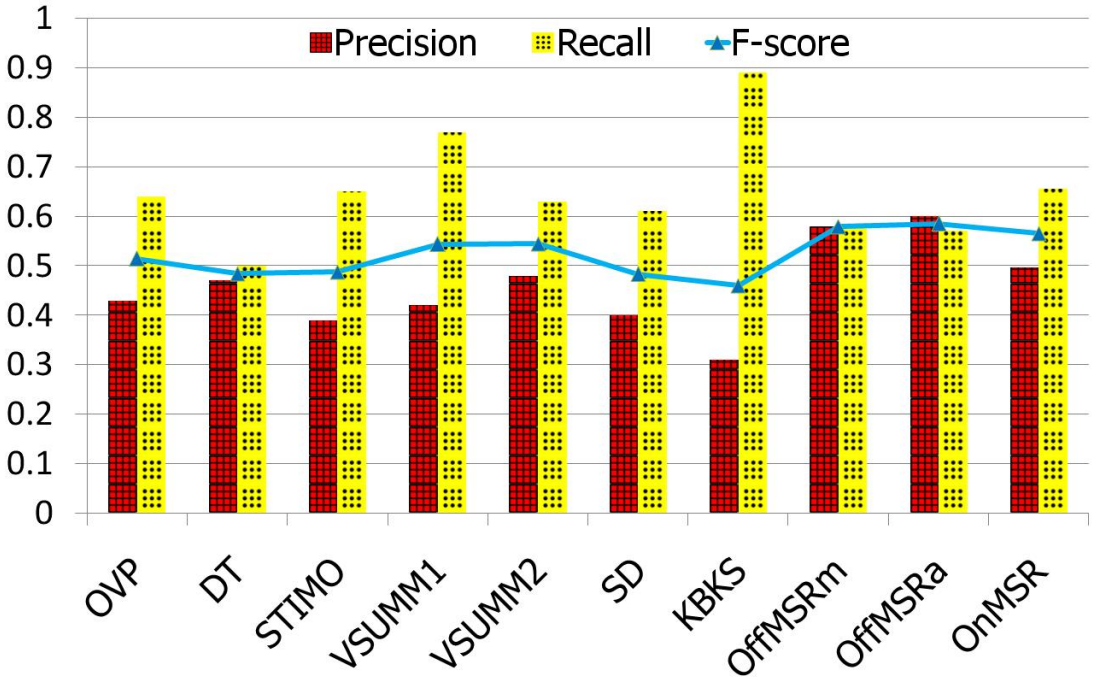


Figure 5.3: Quantitative evaluation in terms of Precision, Recall and F-score for the first dataset.

proposed MSR based video summarization approach is compared with sparse dictionary (SD) based approach [56], VSUMM [59], Open Video Project storyboard (OVP) [60], Delaunay Clustering (DT) [61], STIMO [62], and Keypoint-Based Keyframe Selection [33] (KBKS). The results of OVP, DT, STIMO, and VSUMM from the same website are adopted. In the KBKS algorithm, the keyframes are selected automatically when its STOP coverage is 85%. In the SD algorithm, frames corresponding to the top 10 local maximums are selected firstly according to the importance curve and then a temporal constraint is imposed to further remove the keyframes near to each other. In order to perform fair comparison, T_{POR} is set to 85% in the proposed MSR based video summarization algorithms.

According to the definition of these three quantitative metrics in Section 4.4.2,

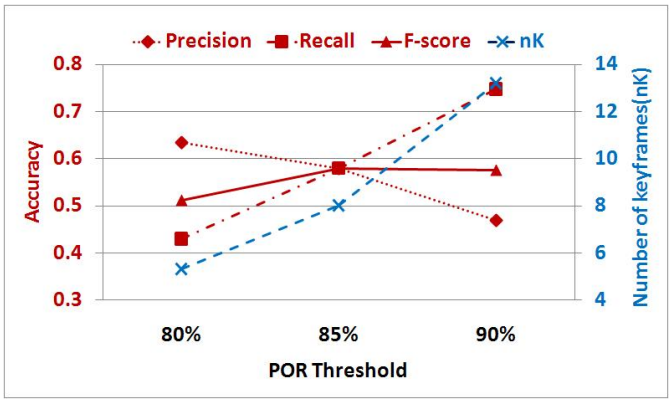
Table 5.2: Performance of each method for the first dataset.

	Precision	Recall	F-score	Average nK
OVP	43%	64%	51.4%	9.66
DT	47%	50%	48.5%	6.2
STIMO	39%	65%	48.8%	9.96
VSUMM1	42%	77%	54.4%	9.62
VSUMM2	48%	63%	54.5%	7.7
SD	40%	61%	48.3%	**10**
KBKS	31%	89%	46.0%	15
OffMSRm	58%	58%	58.0%	8.0
OffMSRa	60%	57%	58.5%	7.62
OnMSR	50%	66%	56.9%	10.58

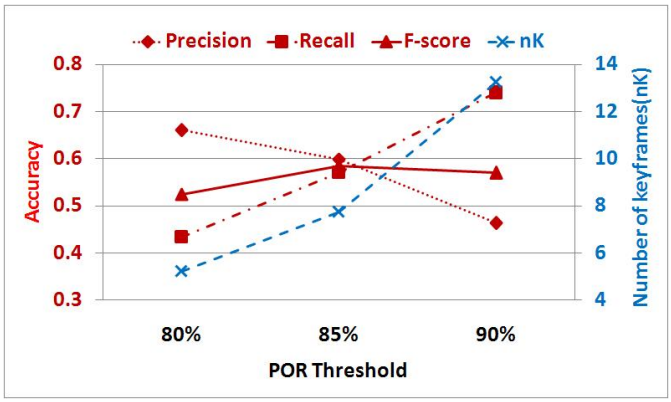
'Precision' reveals the ability to select matched keyframes over all algorithm selecting keyframes, while 'Recall' reflects the ability to select matched keyframes over all ground-truth keyframes. 'F-score' balances these two metrics and evaluates overall performance of summarizing videos. Precision, Recall, and F-score for the first dataset are shown in Fig. 5.3 and detailed in Table 5.2. It is observed that our proposed MSR based video summarization algorithms, including both Off-line version and On-line version, obviously achieve the best performance among all compared methods. Generally, for fewer keyframes that have been selected, fewer keyframes are likely to be matched, and thus resulting a lower Precision. On the contrary, for more keyframes that have been selected, more keyframes are likely to be matched, and thus resulting higher Recall. According to the average number of selected keyframes in these algorithms shown in Table 5.3, although our proposed OffMSR based video summarization algorithms selects fewer keyframes than OVP, STIMO, VSUMM1, and SD, and similar number of keyframes as DT and VSUMM2, the precisions of our proposed OffMSR algorithms are much higher than these two video summarization algorithms, indicating that our proposed OffMSR based video summarization algorithms extract keyframes with much

higher accuracy. In addition, the Recall of our proposed OffMSR based video summarization algorithms is similar to that of most video summarization algorithms considered. Thus, our proposed OffMSR based video summarization algorithms outperform all the methods compared. In addition, the performance of the OffMSRa algorithm is slightly better than the OffMSRm algorithm, indicating that the most average frame (the frame closest to the average of all the frames) is a better choice for initialization than the maximum frame (the frame with the largest magnitude). The performance of our proposed OnMSR algorithm decreases a little compared with the OffMSR algorithms, however, it still outperforms all compared methods. This is because the on-line implementation selects keyframes in a local point of view, which determines whether a video frame is a keyframe or not according all the video frames acquired previously. In addition, when the same POR threshold is adopted, the OnMSR algorithm extracts more keyframes than the OffMSR algorithms.

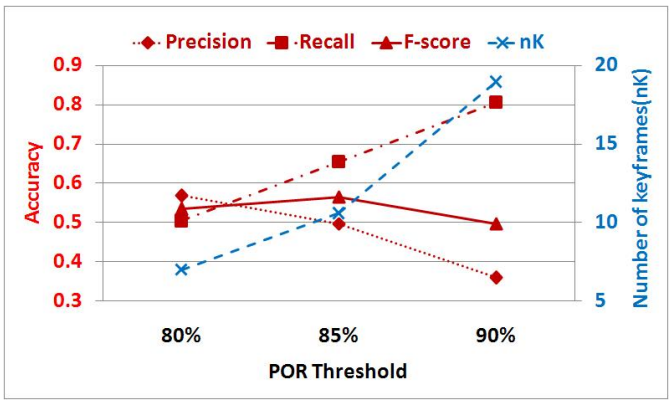
The results of our proposed MSR based video summarization algorithms with different POR thresholds are shown in Fig. 5.4. It is observed that, when the POR threshold increases, more keyframes are selected by the proposed MSR based video summarization algorithms and the precision decreases monotonously. However, the F-score of our proposed approach does not decrease a lot since the Recall increases monotonously. As demonstrated in Fig. 5.4, when $T_{POR} = 90\%$ and more keyframes are selected, our proposed MSR based video summarization algorithms still outperform all compared algorithms. It is also observed that when $T_{POR} = 80\%$ and fewer keyframes are selected, our proposed MSR based video summarization algorithms also outperform all compared video summarization algorithms even if much fewer keyframes are selected, indicating that, our proposed MSR based video summarization algorithms can provide very concise but highly effective summarization results. In addition, the performance of the OnMSR algorithm decreases more than that of the OffMSR algorithm since many



(a) h



(b) h



(c) h

Figure 5.4: The quantitative performance of the proposed MSR based video summarization algorithms with different POR Thresholds: (a). OffMSRm, (b). OffMSRa, and (c). OnMSR.

more keyframes are selected. However, its performance is still better than those of all the other video summarization algorithms compared. Therefore, our proposed OnMSR

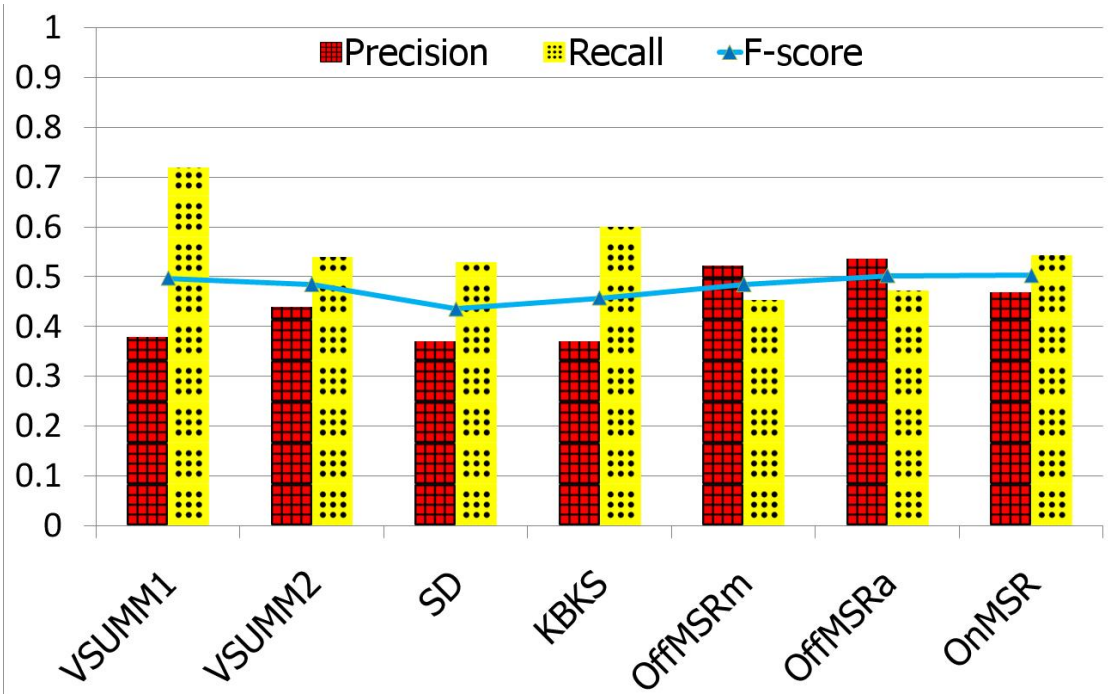


Figure 5.5: Quantitative evaluation in terms of Precision, Recall and F-score for the Second dataset.

Table 5.3: Performance of each methods for the second dataset.

	Precision	Recall	F-score	Average nK
VSUMM1	38%	72%	49.7%	10.26
VSUMM2	44%	54%	48.5%	7.2
SD	37%	53%	43.6%	8
KBKS	37%	60%	45.5%	11.4
OffMSRm	52%	45%	48.2%	8.12
OffMSRa	54%	47%	50.2%	8.14
OnMSR	47%	54%	50.2%	10.96

algorithm is an effective on-line video summarization algorithm.

The second dataset with human-selected ground-truth keyframes from five users is also available at the VSUMM [59] official website. The results of VSUMM from the same website are adopted for comparison. In addition, the SD algorithm [56] and the KBKS algorithm [33] are also considered. The experimental settings for all the algorithms are similar to those in the first dataset. The experimental results of our

Figure 5.6: Sample video summarization results of all the methods for the 5th video of the first dataset, from top to bottom: OVP, DT, STIMO, VSUMM1, VSUMM1, SD, KBKS, OffMSRm, OffMSRa, and OnMSR.

proposed MSR based video summarization algorithms, together with that of VSUMM, SD, and KBKS, are shown in Table 5.3. Although the performance of the proposed OffMSRm is slightly lower than those of the VSUMM algorithms, our another off-line version, the OffMSRa algorithm, clearly outperforms all the video summarization algorithms considered. This also confirms that the frame closest to the average of all the frames is a better choice for initialization. It should also be noted that our proposed on-line version of MSR based algorithm outperforms all considered video summarization methods, including its corresponding off-line version, although the dataset becomes a little difficult to summarize and the performance of all the algorithms decreases a lot.

5.4.2 Case Studies

This section presents several summarization samples for subjective evaluation. The first example is from the 5th video in the first dataset. The video summarization results of all considered video summarization methods are shown in Fig 5.6, in which all the methods to be compared are shown in each row. Being a human selected summary, OVP is able to roughly cover the content, but there are still some details missing. Although DT selects fewer keyframes than others, its selected keyframes are similar to each other, especially for the two in red rectangular. Being a clustering-based method, STIMO fails to extract all clusters due to large intraclass and low interclass visual variance. VSUMM improves clustering-based method by eliminating meaningless frames before clustering and similar keyframes after clustering. In addition, a keycluster selection is involved in VSUMM2 [59]. Thus, VSUMM1 and VSUMM2 achieve better results than the other compared video summarization algorithms. Since sparse relaxation in SD cannot guarantee real sparsity, it selects three keyframes about airplanes. Although KBKS algorithm selects more keyframes in average than others according to Table 5.2, it selects fewer keyframes in this video to provide a very concise summarization. Compared with all these video summarization algorithms, our proposed OffMSR algorithms, including OffMSRm and OffMSRa, top the coverage of contents without redundancy, though they have slightly different results. Although VSUMM provides similar results with our OffMSR, the keyframes being selected by OffMSR contains more details than that by VSUMM. For example, the airplane keyframe selected in OffMSR provides more details about the airplain. In addition, VSUMM misses the last keyframe in our proposed OffMSR algorithms. Our proposed OnMSR selects several keyframes of the airplane, resulting similar results. This is because OnMSR needs more keyframe to reduce REs of coming keyframes in the beginning of video. Therefore, as listed in Table 5.2, OnMSR generally selects more keyrames than its corresponding Off-line versions.

Figure 5.7: Sample video summarization results of all the methods for the 49th video of the second dataset, from top to bottom: VSUMM1, VSUMM1, SD, KBKS, OffMSRm, OffMSRa, and OnMSR.

Such part-view nature is a natural shortcoming for most on-line video summarization algorithms since they determine whether the incoming frame is a keyframe or not according to information of previously selected frames only. However, all the keyframes extracted in the OnMSR method are distinctive, even for the several keyframes with the airplane, indicating that it is an effective on-line algorithm for video summarization.

The second example is from the 49th video in the second dataset. The video summarization results of all the methods are shown in Fig 5.7. It is also confirmed that our proposed OffMSR based algorithms top the coverage of contents without redundancy among all the video summarization algorithms. The results of our proposed OnMSR algorithms are not concise enough to represent the video due to its the part-view nature of on-line algorithms.

Figure 5.8: Sample video summarization results of our proposed OffMSRa algorithm when different levels of reconstruction are adopted for the 5th video of the first dataset.

5.4.3 Summarization with Different Lengths

In the proposed MSR based video summarization algorithms, within each scene of a video, the number of selected keyframes is not predefined, but is determined by the level of reconstruction. A static/low-motion scene may produce fewer keyframes with high percentage of reconstruction. On the other hand, scenes with substantial change of contents may require more keyframes to reach the same level of reconstruction. Therefore, we could adjust the length of the summary by tuning the level of reconstruction, which is more intuitive than setting a fixed number, and is more adaptive to different kinds of videos. This flexibility is mostly lacked in the previous works. The quantitative performance of our proposed algorithms with different POR thresholds for adaptive summarization has been explained as shown in Fig. 5.4. Therefore, in this section, visual results for adaptive summarization are presented for subjective evaluation.

As explained in Fig. 5.1, the proposed OffMSR algorithm extracts keyframes iteratively until it reaches the predefines level of reconstruction. Fig. 5.8 lists the summarization results of our proposed OffMSRa algorithm when different POR thresholds are adopted for the 5th video of the first dataset. When the level of reconstruction is set to 80%, only six keyframes are extracted, resulting in a low-level video summarization. When the level of reconstruction grows up to 85%, three more keyframes are extracted.

Figure 5.9: Sample video summarization results of our proposed OnMSR algorithm when 80% is adopted as the level of reconstruction for the 25th video.

Figure 5.10: Sample video summarization results of our proposed OnMSR algorithm when 90% is adopted as the level of reconstruction for the 25th video.

Moveover, six more keyframes are extracted for 90% of reconstruction, indicating a more detailed coverage of the video. That is, the proposed OffMSRa algorithm can summarize videos into different lengths. Therefore, it is possible to further customize the granularity of the summary for each scene automatically or via user interaction. Similar conclusion can also be drawn in the proposed OffMSRm algorithm.

As shown in Fig. 5.2, the proposed OnMSR algorithm determines whether a frame is a keyframe or not when it is acquired. Therefore, when different POR thresholds are adopted, different summarization results will be obtained. The video summarization results of the proposed OnMSR algorithm when 80% and 90% is adopted as POR thresholds for the 5th video of the first dataset is shown in Fig. 5.9 and Fig. 5.10, respectively. Compared with the video summarization results of 85% shown in Fig. 5.6, when POR threshold increases, a high-level summarization with more frames will be

produced and more details will be kept in the video summarization results. However, as shown in Fig. 5.10, if POR threshold is set too high, transitional frames will be selected as keyframes, especially the blurred transitional frames labeled by red rectangular. In a video, transitional frames are acquired before next distinguished frames. Therefore, in on-line applications, when we try to summarize video with high-level reconstruction, transitional frames are more likely to be selected as keyframes and and thus affect the summarization results. Therefore, special consideration should be taken to handle these blurred transitional frames in on-line applications.

5.5 Summary

In this chapter, an L_0 constrained MSR model is presented for video summarization by reconstructing all the frames in a video as accurate as possible with as few frames as possible. Specifically, an off-line solution and an on-line solution are proposed by utilizing the proposed MSR based video summarization principle. In addition, a POR criterion is proposed to intuitively guide users in selecting an appropriate length of the summary. Experimental results demonstrate that our MSR based video summarization approaches outperform state-of-the-art ones and are flexible to generate various lengths of keyframe based summaries. Moreover, our MSR based video summarization approaches can be utilized to summarize not only the structured videos, such as news, sports, or surveillance videos, but also the consumer videos without pre-defined structures. One of the interests in our future work is to further extend the proposed method for image summarization in on-line shopping and on-line image recommendation applications [63][64].

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

$L_{2,0}$ Constrained Sparse Dictionary Selection for Video Summarization

This chapter proposes an enhanced video summarization approach to the previous chapter based on constrained Sparse Dictionary Selection. It is based on results and includes text that have been published in [65].

6.1 Introduction

Recent developments on sparse dictionary selection have demonstrated promising results to solve these problems for video summarization [56]. In this algorithm, video summarization is performed by extracting a sparse dictionary from the entire video frame pool since keyframes are viewed as a dictionary that can recover the whole video frames without significant information loss. However, the relaxation method to solve this problem, in which an $L_{2,1}$ norm is imposed to enforce the selected key frames as sparse as possible, cannot ensure the sparsity of the reconstruction coefficients directly. As a result, the selected keyframes are not the optimal subset of the video frames that

can reconstruct the original video as accurate as possible, since the contribution of non-keyframes for reconstruction cannot be strictly confined to zero.

Therefore, in this chapter, the $L_{2,0}$ norm is adopted to formulate a simultaneous sparse constraint for dictionary selection based video summarization. As a result, an $L_{2,0}$ based sparse dictionary selection model is proposed for video summarization problems. Moreover, a simultaneous orthogonal matching pursuit (SOMP) based keyframe extraction algorithm is proposed to get an approximate solution for the proposed $L_{2,0}$ based problem directly without smoothing the penalty function. Thus, the contribution of non-keyframes for reconstruction is eliminated by strictly confining the reconstruction coefficients of non-keyframes to zero. Finally, experiments on different video datasets are conducted to demonstrate the effectiveness of the proposed algorithm.

6.2 $L_{2,0}$ Based Dictionary Selection Model for Video Summarization

Since keyframes are concise yet representative to an original video, all the important content of the video should be included in these keyframes. Therefore, given an initial candidate pool $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n] \in I\!R^{d\times n}$, where each column vector $\mathbf{f}_i \in I\!R^d$ denotes a video frame represented as feature vector, the goal of video summarization is to find an subset $\mathbf{F}_K = [\mathbf{f}_{i_1}, \mathbf{f}_{i_2}, \ldots, \mathbf{f}_{i_p}] \in I\!R^{d\times p}$ where $i_1, i_2, \ldots, i_p \in \{1, 2, \ldots, n\}$ such that it can represent all the important information in the video frame set $\mathbf{F}$ without significant information loss:

$$\min_{\mathbf{F}_K} |\mathscr{I}(\mathbf{F}) - \mathscr{I}(\mathbf{F}_k)| < \delta, \tag{6.1}$$

in which $\mathscr{I}(\cdot)$ represents the information implying in a frame set and $\delta > 0$ is the information loss tolerance. Therefore, if the keyframe set is viewed as a dictionary that can reconstruct all the frames in the video, video summarization will be performed

by selecting a dictionary from the video frames such that all the video frames can be reconstructed by such a dictionary:

$$\min_{\mathbf{F}_K} : \|\mathbf{F} - \mathbf{F}_K\mathbf{A}\|_F < \delta, \tag{6.2}$$

in which $\mathbf{A} \in \mathbb{R}^{p\times n}$ is the reconstruction coefficient of $\mathbf{F}$ by $\mathbf{F}_K$, $\|\cdot\|_F$ represents the Frobenius norm of a matrix. Since keyframes are just a subset of frames in the video, video summarization can be viewed as a sparse reconstruction problem that all the frames in the video can be reconstructed by a part of them, which can be formulated as follows:

$$\min_{\mathbf{X}} : \|\mathbf{F} - \mathbf{F}\mathbf{X}\|_F < \delta, \tag{6.3}$$

in which $\mathbf{X} \in \mathbb{R}^{n\times n}$ represents the sparse reconstruction coefficient. Therefore, the reconstruction coefficient $\mathbf{A}$ by the keyframes corresponds to all the nonzero rows of $\mathbf{X}$, which implies that, if the i-th $(i = 1,2,\ldots,n)$ row of $\mathbf{X}$ is not a zero vector, its corresponding i-th frame involves in the reconstruction of other frames and will certainly be selected as a keyframe; Otherwise, it will not be selected as a keyframe.

In this chapter, the $L_{2,0}$ norm is adopted to guarantee the simultaneous sparsity in the reconstruction coefficient $\mathbf{X}$. The $L_{2,0}$ norm of $\mathbf{X}$, which is indeed a general version of L_0 norm that counts the number of zero elements in a vector, is defined as $\|\mathbf{X}\|_{2,0} = L_0(\|X_{i\cdot}\|_2)$, where $X_{i\cdot}$ denotes the i-th row of $\mathbf{X}$. If we construct a new vector $\mathbf{x} \in \mathbb{R}^n$ with $x_i = \|X_{i\cdot}\|_2$, the $L_{2,0}$ norm of $\mathbf{X}$ equivalents to $\|\mathbf{x}\|_0$. Therefore, $L_{2,0}$ norm actually counts the number of zero rows in $\mathbf{X}$ and thus guarantees the simultaneous sparsity. Generally, the number of keyframes should be as small as possible since the keyframes are just a concise representation of the video. Therefore, in this chapter, an $L_{2,0}$ norm based sparse dictionary selection (SDS) model is constructed for video

summarization:

$$\begin{aligned} &\min_{\mathbf{X}} : \|\mathbf{X}\|_{2,0} \\ &s.t. \quad \|\mathbf{F} - \mathbf{F}\mathbf{X}\|_F < \delta. \end{aligned} \tag{6.4}$$

In summary, in the proposed $L_{2,0}$ norm based SDS model, video summarization is performed with the following two criteria: 1) the entire video frames can be reconstructed by the keyframes as accurate as possible, and 2) the size of keyframes is as small as possible.

6.3 Proposed Video Summarization Method

Generally, there are two kinds of methods to solve the simultaneous sparse problems: Greedy Algorithm (GA) [66] and Convex Relaxation algorithm [67]. To the best of our knowledge, only convex relaxation method has been utilized to solve video summarization by adopting $L_{2,1}$ norm to ensure simultaneous sparsity [56]. Little attention has been paid to the use of GA for video summarization. GA follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. Although it could not produce an optimal solution for some problems, it is able to yield locally optimal solutions that approximate a global optimal solution in reasonable time. On the other hand, the advantages brought by GA such as the low computational complexity of solving the optimization problems containing non-smooth terms and that it can get an approximate solution for the $L_{2,0}$ problem directly without smoothing the penalty function are attractive. Therefore, in this chapter, GA algorithms are adopted to solve the $L_{2,0}$ based SDS model for video summarization.

Orthogonal Matching Pursuit (OMP) [68], a typical GA, has been widely utilized to

solve the L_0 based sparse problem [69]. It has also been extended to solve the simultaneous sparse approximation by a simultaneous OMP (SOMP) algorithm [66]. Although the simultaneous sparse problem in (6.4) is slightly different from traditional simultaneous sparse approximation problem [66], it pursues simultaneous sparsity in coefficient matrix similar to that in simultaneous sparse approximation problem. Therefore, the SOMP algorithm in [66] is extended to solve the $L_{2,0}$ norm based SDS model defined by (6.4) for video summarization.

First, we give a formal description of the proposed SOMP based algorithm for video summarization, and then discuss some of its basic properties. According to the SOMP algorithm in [66], an SOMP based video summarization algorithm is proposed as follows:

Algorithm 3: The proposed SOMP based video summarization algorithm

Input: the whole video frame set $\mathbf{F} \in I\!R^{d\times n}$.
Output: the summarized keyframe set $\mathbf{F}_K \in I\!R^{d\times p}$.
Initialization:
Initialize the residual matrix $\mathbf{R}_0 = \mathbf{F}$, the index set $\Lambda_0 = \varnothing$, and the iteration counter $t = 0$.
SOMP iteration for video summarization:
While the termination condition is not satisfied:
1). Update iteration: $t \leftarrow t+1$.
2). Find an index λ_t that the reconstruction error of the current residual $\mathbf{R}_0$ by its corresponding frame $\mathbf{f}_{\lambda_t}$ is minimized:

$$\lambda_t \leftarrow \arg\min_{1\le k\le n} \|\mathbf{R}_{t-1} - \mathbf{f}_k \mathbf{A}_k^{t-1}\|_F, \tag{6.5}$$

where $\mathbf{A}_k^{t-1}$ represents the reconstruction coefficient of $\mathbf{R}_{t-1}$ by $\mathbf{f}_k$.
3). Update the index set: $\Lambda_t \leftarrow \Lambda_{t-1} \cup \lambda_t$.
4). Update keyframe set: $\mathbf{F}_t^K = \mathbf{F}_{\Lambda_t}$.
5). Update the reconstruction coefficient:

$$\mathbf{X}_t \leftarrow \arg\min_{\mathbf{X}} \|\mathbf{F} - \mathbf{F}_K^t \mathbf{X}\|_F. \tag{6.6}$$

6). Update the residual:

$$\mathbf{R}_t = \mathbf{F} - \mathbf{F}_K^t \mathbf{X}^t. \tag{6.7}$$

End **While**

In the proposed SOMP algorithm for video summarization, three issues must be

solved before its implementation: how to solve the minimization problem defined by (6.5) to obtain the reconstruction coefficient $\mathbf{A}_k^t$, how to solve the minimization problem defined by (6.6) to obtain the reconstruction coefficient $\mathbf{X}_t$, and how to stop the iteration.

According to the minimization problem defined by (6.5), a frame that can best reconstruct the current residuals is selected as a new keyframe at each iteration of the SOMP algorithm. Therefore, the reconstruction coefficient for $\mathbf{R}_t$ by $\mathbf{f}_k$, denoted by $\mathbf{A}_k^t$, must be predetermined. In order to assess the ability of the k-th frame to reconstruct the current residuals, its corresponding reconstruction coefficient $\mathbf{A}_k^t$ is defined as follows:

$$\mathbf{A}_k^t = (\mathbf{f}_k^T \mathbf{f}_k)^{-1} \mathbf{f}_k^T \mathbf{R}_t = \frac{\mathbf{f}_k^T \mathbf{R}_t}{P(\mathbf{f}_k)}, \tag{6.8}$$

in which $P(\mathbf{f}_k) = \|\mathbf{f}_k\|_2$ is the L_2 norm of the k-th frame.

In order to characterize the ability of reconstruction by the current keyframe set, a minimization problem defined by (6.5) is formulated to calculate the current reconstruction coefficient. In the proposed SOMP algorithm, we must guarantee that a frame cannot be selected as a keyframe for several times. Therefore, the residuals must be irrelevant to the keyframes. As a result, the orthogonal subspace projection (OSP) is adopted to calculate the current reconstruction coefficient $\mathbf{X}_t$:

$$\mathbf{X}_t = ((\mathbf{F}_K^t)^T \mathbf{F}_K^t)^{-1} (\mathbf{F}_K^t)^T \mathbf{F}. \tag{6.9}$$

Generally, it is difficult to determine how many keyframes should be selected [33] in a video summarization problem. This is still an open issue to select a reasonable number of keyframes without prior knowledge. One of the plausible ways to solve this problem is to predefine how many keyframes should be extracted by popular intrinsic dimensionality (ID) estimation algorithms, such as principle component analysis (PCA), an information theoretic criterion (AIC), and minimum description length (MDL) [70]. A

more reasonable solution is to set an adjustable stopping criterion which can be tuned to summarize the video frames with different length for different applications. Therefore, in this chapter, a percentage of residuals (POR) threshold, which measures the energy of residuals that cannot be reconstructed by current keyframe set, is proposed to perform flexible video summarization. The POR at t-th iteration in the proposed SOMP algorithm is defined as follows:

$$POR_t = \frac{E(\mathbf{R}_t)}{E(\mathbf{F})} = \frac{\|\mathbf{R}_t\|_F}{\|\mathbf{F}\|_F}. \tag{6.10}$$

Therefore, if POR_t decreases below a predefined threshold T_{POR}, the iteration will stop and all the keyframes have been extracted; Otherwise, more keyframes are required to be selected. Consequently, the video summarization results are tuned by T_{POR} for different values to summarize the video frames with different degrees of reconstruction, and thus, summarizing video frames with different lengths for different levels of summarization.

6.4 Experiments and Discussions

In this section, two popular video datasets are adopted to quantitatively evaluate the performance of our proposed SOMP based video summarization algorithm. The experimental settings and evaluation metrics have been introduced in Section 4.4.1 and Section 4.4.2 respectively.

6.4.1 Performance Evaluation

The first dataset with human-selected ground-truth keyframes from five users is available at the VSUMM [59] official website. When calculating quantitative metrics, the average of the results among the five ground-truth sets of keyframes is adopted. Our

Table 6.1: Performance of each methods for the first dataset.

	Precision	Recall	F-score
OVP	43%	64%	51.4%
DT	47%	50%	48.5%
STIMO	39%	65%	48.8%
VSUMM1	42%	77%	54.4%
VSUMM2	48%	63%	54.5%
SD	40%	61%	48.3%
KBKS	31%	89%	46.0%
SOMP	55%	73%	63.1%

Table 6.2: The average number of selected keyframes for the first dataset.

OVP	DT	STIMO	VSUMM1	VSUMM2	SD	KBKS	SOMP
9.66	6.2	9.96	9.92	7.7	10	15	11.16

proposed SOMP based video summarization approach is compared with sparse dictionary (SD) based approach [56], VSUMM [59], Open Video Project storyboard (OVP) [60], Delaunay Clustering (DT) [61], STIMO [62], and Keypoint-Based Keyframe Selection [33] (KBKS). The results of OVP, DT, STIMO, and VSUMM from the same website is adopted. In the KBKS algorithm, the keyframes are selected automatically when its STOP coverage is 85%. The top 10 frames selected by SD based method are used for evaluation to comply with its experimental settings, since the average number of ground truth keyframes is around 10. In the proposed SOMP algorithm, the 360-D feature vector utilized in [56] is adopted to represent each frame in a video. In order to perform fair comparison, T_{POR} is set to 20% in the proposed SOMP based video summarization algorithm since it extracts around 10 keyframes.

According to the precision, recall, and F-score shown in Table 6.1, obviously, our proposed SOMP based video summarization algorithm achieves the best performance among all compared methods. By further analyzing the selected keyframes in terms of the average number of selected keyframes shown in Table 6.2, it is noticed that DT

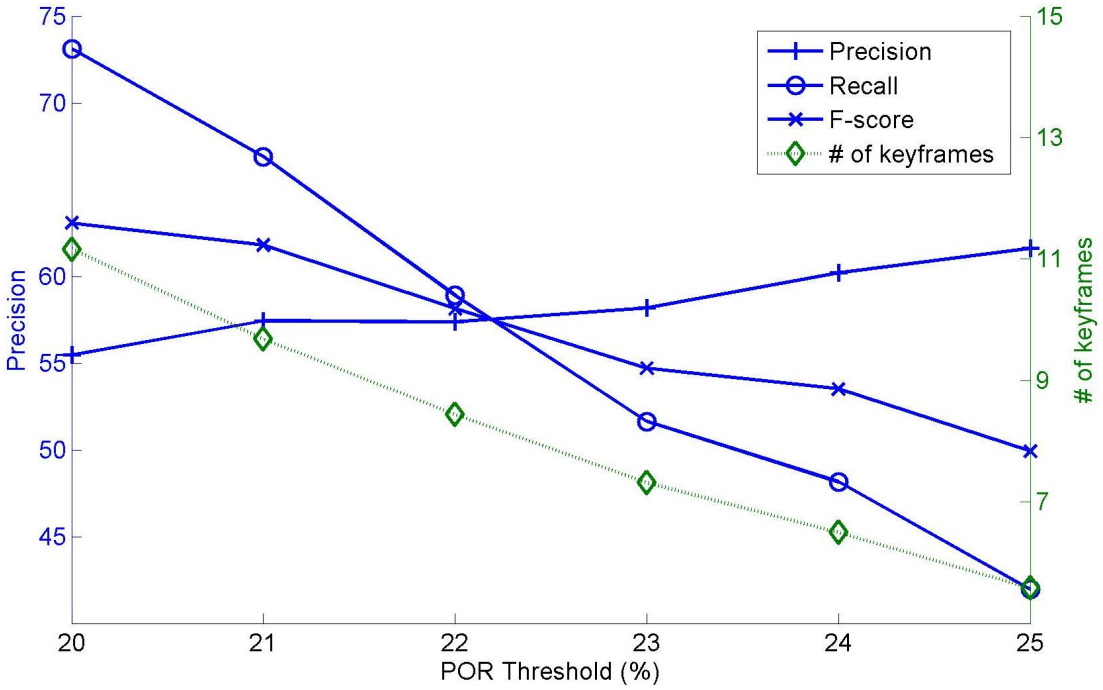


Figure 6.1: The performance of our proposed SOMP based video summarization algorithm for different POR thresholds.

and VSUMM2 select fewer keyframes than others, so they have slightly higher precision values and lower recall rates due to the limited selection of keyframes. Although our proposed SOMP produces similar numbers of keyframes on average with OVP, STIMO, VSUMM1, and SD, its performance in terms of precision and F-score clearly outperforms that of these four algorithms. Our proposed SOMP based video summarization approach is inferior to the KBKS algorithm in terms of Recall since it selects fewer keyframes. However, in terms of Precision and F-score, our proposed approach obviously outperforms the KBKS algorithm.

The results of our proposed SOMP with different POR thresholds are shown in Fig. 6.1. When the POR threshold increases, fewer keyframes are selected by the proposed approach. However, the F-score of our proposed approach does not decrease a lot. Even when $T_{POR} = 25\%$ and only an average of 5.58 keyframes is selected, our proposed approach still outperforms other considered video summarization approaches.

Table 6.3: Performance of each methods for the second dataset.

	Precision	Recall	F-score
VSUMM1	38%	72%	49.7%
VSUMM2	44%	54%	48.5%
SD	37%	53%	43.6%
KBKS	37%	60%	45.5%
SOMP	46%	61%	52.5%

Therefore, the proposed SOMP algorithm can provide very concise but high-effective summarization results, which is extremely useful when the amount of video content is increasing rapidly.

The second dataset with human-selected ground-truth keyframes from five users is also available at the VSUMM [59] official website. The results of VSUMM from the same website is adopted for comparison. In addition, the SD algorithm [56] and the KBKS algorithm [33] are also considered. The experimental settings for all the algorithms are similar to those in the first dataset. The experimental results of our proposed SOMP based video summarization algorithm, together with those of VSUMM, SD, and KBKS, are shown in Table 6.3. It is also confirmed that our proposed approach outperforms all other methods, although the dataset becomes a little difficult to summarize and the performance of all the algorithms decreases a lot.

6.4.2 Case Studies

In this section, the 4th video of the first dataset is selected for subjective evaluation. The summarization results by all the considered video summarization methods are listed in Fig 6.2. As illustrated, DT tends to select fewer keyframes than others. In addition, since DT and STIMO are clustering-based methods, they normally select the keyframes representing average information within a cluster (i.e., close to the centroid of a cluster), which might cause problems when a cluster contains frames from multiple scenes. In that case, it will likely select keyframes during a fading transition that are most similar

Figure 6.2: Sample video summarization results of all the methods for the 4th video in the first dataset, from top to bottom: OVP, DT, STIMO, VSUMM1, VSUMM2, SD, KBKS, and SOMP(ours). The frames in red rectangular are possible redundant frames.

to all the frames in the cluster, as clearly shown in Fig 6.2. This problem of DT and STIMO has also been observed from many other videos in our experiments. Being a human selected summary, OVP is able to roughly cover the content, but there are still some details missing. Although SD utilizes the same feature with our proposed approach, it loses much details in this case, and gives some redundant keyframes as pointed out in the red rectangular in Fig 6.2. Comparing our methods with KBKS and VSUMM, we notice that they cover similar amount of details of the video, while our methods top the coverage of contents. Although similar keyframes (shown enclosed by a red rectangular in Fig 6.2) emerge in our approach, they are selected just because we want to obtain more information of the video by selecting more keyframes. The summarization results by the proposed SOMP based approach with different POR thresholds are also indicated out in this figure. Combined with the quantitative evaluation in Fig. 6.1, our proposed SOMP based approach is an 'always-in-focus' algorithm which can summarize video

sequence with different lengths for different applications with high accuracy.

6.5 Summary

In this chapter, an $L_{2,0}$ constrained sparse dictionary selection model is constructed for video summarization. Specifically, an SOMP based keyframe extraction algorithm is proposed to obtain an approximate solution for the proposed $L_{2,0}$ based problem directly without smoothing the penalty function. In addition, a percentage of residuals (POR) threshold is utilized to provide flexibility for different kinds of videos by summarizing video sequence with different length. The experimental results demonstrate that our proposed approach outperforms the state-of-the-art and is flexible to generate various lengths of keyframe based summaries. Moreover, the proposed SOMP based video summarization approach can be utilized to summarize not only the structured videos, such as news, sports, or surveillance videos, but also the consumer videos without pre-defined structures. One of the interests in our future work is to further extend the proposed method for personal photo summarization.

Chapter 7

Towards Multimedia Summarization - StoryImaging: from Text Story to Images

This chapter introduces a novel system to achieve multimedia storytelling. It is based on results and includes text that have been published in [71].

7.1 Introduction

Video summarization can be regarded as a subset of multimedia content summarization. Multi-modal content analysis will become the next hot topic, which integrates the information of textual, audio and visual features of a piece of content to generate a more meaningful and enriched representation. This is especially beneficial for the presentation of a large piece of content (e.g., a long video or a long text story) to users. Since this topic requires many content analysis techniques involved, we firstly make a pioneering step towards this ultimate goal with visualization support to textual stories.

Visual information is essential for our daily lives. As indicated in [72], 80% of

human cognition is obtained from visual information. For example, some natural languages are historically derived from pictographs, such as Chinese and Egyptian, as evidenced with ancient cave paintings. In addition, visual information generally contains rich content and is very expressive. As said, one picture is worth a thousand words. There are plenty of such examples. Journalists often use several images to help illustrate news stories and a movie is a visual illustration of a novel (with very high cost). Images are also useful for people with reading difficulties. In the web era, emoticons are developed to enhance communication experiences in Instant Messengers. However, we have to deal with a large amount of textual information every day, such as news articles and documentaries. It is very often that we hope to know what the person looks like and what the place is in reading a news article even when knowing that a phrase represents a person's name or a location. Furthermore, different people will have different interpretation of the same text due to their diverse background (e.g. culture and knowledge). Besides this, diverse background may lead to mistaken interpretation of a textual story without the help of visual aids. All these factors increase the difficulty to human beings in comprehending textual information. Therefore, it would be ideal to augment textual information with suitable images so as to facilitate comprehension in consuming textual documents.

To enrich an arbitrarily given textual story (e.g. a news article or a biography), a large scale repository with images and accompanying text is required. Nowadays the Web has become the largest database on the planet and commercial search engines allow us to conveniently access a large number of images by expressing information needs with a number of words. However, several issues should be addressed when search engines are utilized for such purpose. Firstly, the search results returned by current commercial search engines are generally too noisy, which demands further interaction (e.g. refinement) from users. Secondly, users have to conduct some queries during the

reading, if there are a number of unknown items such as persons and locations, which is time consuming and significantly interrupts information consumption. Thirdly, it is difficult for users to gain an overview of the story, since results returned by a series of independent queries are isolated.

Fang [73] investigated the relationship between texts and pictures in children's storybooks and summarized six roles of visual illustration as, 1) establishing the setting, 2) defining/developing the characters, 3) extending/developing the plot, 4) providing a different viewpoint, 5) contributing to the text coherence, and 6) reinforcing the text. While reading a piece of descriptive story, users who (i.e. persons involved in the event), where (i.e. settings such as locations), and what (i.e. activities such as giving a speech). We currently focus on identifying named entities such as persons (who) and locations (where), and key terms (what) as initial queries [74].

Based on the above observations, we propose a novel approach, namely *StoryImaging*, to automatically fetching images from the Web and selecting suitable ones for the visual illustration of a given textual story. As shown in Figure 7.1, our approach consists of five components: text analysis, image search, search result re-ranking, near duplicate removal, and image clustering. Text analysis includes Part-of-Speech (POS) tagging for extracting noun words, identification of named entities, and summarization of the text story for the following re-ranking. Image search is to collect an initial set of images by issuing queries based on named entities and key terms in each sentence. Re-ranking and near duplicate removal is to improve the quality of the returned images. And visual clustering is to enable convenient overview of the story. As shown in Figure 7.2 for a sample excerpt from Wikipedia, users can easily gain an overall picture of the company from its logos, key products, and key persons.

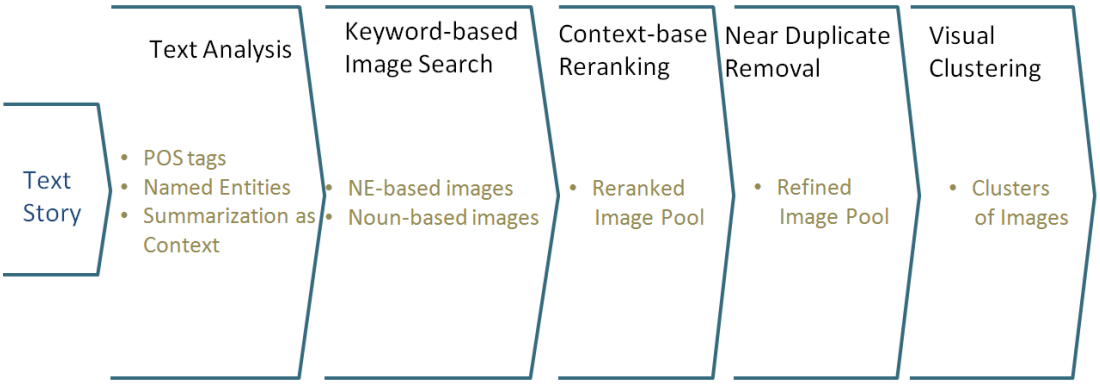


Figure 7.1: Workflow of our *StoryImaging* approach.

7.2 Related Work

To the best of our knowledge, our work is one of the first efforts harvesting web images for story illustration. The most similar work to ours is the Story Picturing Engine (SPE) [75]. It consists of three steps: 1) identifying semantic keywords from input texts; 2) searching a database of annotated images with those keywords for candidate images; and 3) selecting images from the candidate images with mutual-reinforcement-based rank. However, it worked only on a small closed database, and semantic keywords were limited to objects such as flower and car. In addition, the global context of the story was not taken into account. In [76], a system was developed to assist news reading by selecting Flickr images. However, its purpose was to attract and retain users' attention in news reading through matching content with noisy tags, instead of facilitating comprehension without thoroughly analyzing the content of news and images.

Recently, some researchers have also been interested in interpreting words (e.g. objects, persons, and landscapes) with visual information. Word2Image [77] was proposed to produce a set of high quality, precise, diverse and representative images for a given word through correlation analysis and clustering of semantic and visual features. In [78] by Feng *et al.*, interpreting textual inputs with images was treated as a by-product

Figure 7.2: A *StoryImaging* sample.

of image annotation, where global context of the input is discarded. In addition, it was also constrained to a closed set of news archives. In [79], Wang *et al.* proposed to interpret tags with the distribution of visual words obtained through sample images, which is an analogy to traditional dictionary which defines terms with real words. Nevertheless, visual words obtained through clustering local features generally lack semantic meanings. In [80], Deng *et al.* proposed to build ImageNet by following the hierarchy of WordNet. For each noun word in WordNet, relevant web images were manually collected, which is an expensive effort. It is also worth mentioning that the above three studies only focus on individual words, instead of a whole text story conveying more information. In the domain of named entities, Taneva *et al.* proposed to improve image search for persons with an existing knowledge base[81]. In [82], Lu *et al.* proposed to collect landmark images for travelogues. However, these methods generally work on very short text (e.g. several words only), instead of a whole textual story conveying more information.

In summary, our contributions lies in three aspects, 1) working at story level for a wide range of documents such as news articles, documentaries, and Wikipedia entries; 2) applying story context to harvest web images; and 3) clustering-based informative presentation for engaging and explorable user experiences.

7.3 Proposed StoryImaging Approach

Using the context and keywords extracted from text, the *StoryImaging* system harvests images from the web where there are abundant image resources, refines the image pool by context-based reranking and near-duplicate removal, and eventually outputs sentence-by-sentence image lists, which serves as an illustration for users reading the story in the order of sentences, image lists and visual image clusters, which gives an overall picture of the story. As shown in Figure 7.2 for a sample excerpt from

Wikipedia, users can easily gain an overall picture of the company from its logos, key products, and key persons.

7.3.1 Text Summarization

Since our approach allows free user inputs, the input text could vary from a complete story the user is interested in to a snippet of a news article, a section of a document, or a fraction of a web page. Therefore, the position of a sentence cannot be used to determine its saliency, which has been proven to be one of the key features in automatic text summarization [83]. In this case, summarization of a given input is achieved by ranking the saliency of sentences, and extracting the named entities and salient noun words in top sentences [84], which will also be the context for our visual illustration.

First, stop words are removed from the story and each word is lemmatized. Term frequency $tf(w)$ is calculated for each word w. Then inverse document frequency $idf(w)$ is measured using a corpus of one-year TV news transcripts from five channels. Note that this corpus can be replaced by any other general-purpose ones to avoid bias. Next, named entities are extracted from each sentence. For the i-th sentence S_i with its collection of words C_i, its saliency is scored by

$$S_i = \sum_{w \in C_i} tf(w) * idf(w) + |NE \in C_i|, \tag{7.1}$$

where $|NE \in C_i|$ is the number of named entities (NE) in sentence i. The sentences can then be sorted by its saliency, and all named entities in top $Num_{Context}$ sentences are combined as the context of the story. Note that other advanced summarization algorithms can be applied to obtain story context. In our experiments, $Num_{Context}$ is set to 3 empirically.

7.3.2 Keyword-based Image Search

Named Entities (NEs) such as people and organizations are the focus of a story. Therefore, we aim to obtain images by forming queries with the words of those NEs. Named entity based image search first removes duplicate entities by regarding two names identical if one is the prefix or suffix of the other, such as "Microsoft Corporation" vs "Microsoft" and "President Obama" vs "President" and "Obama". For each NE, images are then collected from the commercial search engines such as Bing and Google. Meanwhile, location information in the story is also utilized to generate maps for users so that they know where the related locations are.

Besides named entities, noun words generally reflect the fundamental meaning of a sentence. Therefore, we collect extra images by forming a query with all the noun words of each sentence.

After retrieving images from the searches with their source webpages, each named entity and each sentence have a list of candidate images, denoted by $L(NE)$ and $L(S_i)$, respectively. Each sentence links to several corresponding named entities mentioned within that sentence, so $L(NE \in C_i)$ and $L(S_i)$ are formed to highlight the key $NE \in C_i$ and the content of each sentence.

7.3.3 Context-based Re-ranking

Images collected from search engines are related to the keywords, but are not necessarily relevant to the story itself. Therefore, re-ranking of these candidate images is necessary to obtain the story-relevant ones and to filter the noises.

The surrounding text of an image in a web page implies its visual content, so we use this textual feature to further re-rank the image pool against the story context. For each candidate image I_j, its relevance to the context is measured by calculating the cosine similarity between I_j's surrounding text and the story context based on their

tf-idf vectors. That is:

$$Relevance_j = cos(I_j.SurroundText, StoryContext) \tag{7.2}$$

After the re-ranking of each $L(NE \in C_i)$ and $L(S_i)$, only the top images of $L(NE)$ and $L(S_i)$ will be shown to the user.

We use the summary as the context, rather than the whole story, because a condensed context can effectively define the story background, thereby filter the images with a small but precise funnel. The whole story might have diverse words so the relevance will be affected if irrelevant words are matched with the surrounding text of an image. In addition, we do not add the context into the search query directly, otherwise the search result will be significantly biased by the context words.

7.3.4 Near Duplicate Removal

Duplicates or near duplicates are very common in web search due to the nature of the Web. Therefore, search engines may return different version of the same image content, though significant efforts have been endeavoured for Near-Duplicate Detection. It would be ideal to remove the near duplicated images to maximize the information delivered to users.

In [85], an efficient technique was proposed to quickly identify near duplicate images for a query from a database of tens of thousands of images. We adopt such an algorithm to efficiently discover near duplicates and to enable real-time interaction for users. The algorithm makes use of multi-resolution wavelet decompositions of an image, which are distilled into small *signatures* for fast computing and comparison based

on the *image querying metric*. The *image querying metric* is given by:

$$\begin{aligned} ||Q_c, T_c|| =& w_{0,0}|Q_c[0,0] - T_c[0,0]| \\ &+ \sum_{i,j:\tilde{Q}_c[i,j]\neq 0} w_{bin(i,j)}(\tilde{Q}_c[i,j] \neq \tilde{T}_c[i,j]), \end{aligned} \tag{7.3}$$

where Q_c (T_c) represents a single colour channel c of the wavelet decomposition of the query image (the target image), $Q_c[0,0]$ ($T_c[0,0]$) is the scaling function coefficient corresponding to the overall average intensity of that colour channel of the query image (the target image), $\tilde{Q}_c[i,j]$ ($\tilde{T}_c[i,j]$) represents the $[i,j]$-th truncated and quantized wavelet coefficients of Q (T), and $w_{i,j}$ is the weight for a term grouped into *buckets* by $bin(i,j)$. Therefore, only a small number of weights $w_{i,j}$ need to be determined experimentally.

7.3.5 Visual Clustering

To give users an overall picture of a story, visual clustering is applied to capture various visual groups for easier understanding and better organization of the result. As normal clustering leads to separated groups of images, the semantics of a story are broken into isolated parts. Therefore, we propose to address such problem of clustering algorithms by discovering the minimum spanning tree of the graph formed by resulting clusters. Our extension is based on the Reciprocal Election algorithm proposed in [86] for a lightweight clustering and eventually construct a graph by connecting each cluster with a minimum spanning tree, which can be visualized as a force-directed graph. Details of this algorithm is shown in Algorithm 4. In our approach, the Colour and Edge Directivity Descriptor (CEDD) is employed to characterize visual content by taking both colour and texture attributes into account [40]. Selection of such features is also a trade-off between performance and efficiency.

Algorithm 4: Reciprocal Clustering

Input : Set S containing all candidate images, parameter m to control cluster granularity (set to 5 in our experiments)
Output : A Graph G connecting Clusters C

```
Initialize votes map V[0,...,k] = 0,...,0
for each image i in S do
    Rank S into L_i based on visual similarity to i
    for each image j in L_i do
        V[j] += 1/r, where r is the rank of j in L_i
While V is not empty do
    Let R_i be the item with the highest score in V
    Remove R_i from V
    Initialize new cluster C with representative R_i
    for all items s in V do
        if R_i is in top-m of L_s then
            add s to cluster C
            remove s from V
Initialize Rep[0,...,n] = [null,...null] (n is |C|)
For each cluster C_x from C
    Rep[x] = R_x which is the representative image in C_x
Construct a minimum spanning tree G from Rep
```

7.4 Demonstration

A web based demo system has been developed for the proposed approach, and the frontend screenshot is presented at Figure 7.3 and Figure 7.4.

7.5 Summary

We present a novel framework to automatically enrich a given textual story by informatively organizing relevant web images. Based on such framework, we develop a simple yet effective and efficient *StoryImaging* system. It firstly identifies key terms such as named entity and extract a story summary. Then an image pool will be obtained by issuing queries formed with those key terms to commercial search engines and the story summary will be utilized to re-rank image candidates. At last, images clustered in terms

Figure 7.3: Frontend of StoryImaging system.

Figure 7.4: Frontend of StoryImaging system.

of visual features will be presented to users. Since our approach utilizes web images, eventually any textual story can be illustrated visually.

In our future work, we will extend the current framework to further integrate other modalities such as video and audio in augmenting textual information so as to eventually achieve multimedia storytelling [87].

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 8

Conclusion and Future Work

This thesis investigates the task of automatic video summarization from three novel perspectives, including a local keypoint based top-down approach, formation of the task to sparse dictionary reconstruction, and a text-to-image system for multimedia summarization. This chapter summarizes the main contributions and suggests directions for future work.

8.1 Main Contributions

8.1.1 Local Keypoint based Keyframe Selection

By utilizing the discriminative power of local visual keypoints, a novel keypoint-based keyframe selection method is proposed in Chapter 3. It starts by building a global pool of unique local visual keypoints from all video frames. After that, representative keyframes are selected by a greedy algorithm based on two criteria, namely Coverage

and Redundancy, to achieve maximum video content coverage and minimum redundancy. The experimental results on both case studies and quantitative evaluation indicate very promising performance compared to other global-visual-feature based methods.

8.1.2 Top-Down Approach for Video Summarization

Chapter 4 presents a top-down approach for video summarization framework, which decompose the problem into two sub-problems: scene identification and scene summarization. Scene identification is achieved by global visual feature clustering based on the fact that frames within a scene are visually similar, while scene summarization is accomplished through selecting keyframes with most representative local visual feature. Users can easily fine tune the summarization results by adjusting the two intuitive metrics *Coverage* and *Redundancy*. The experimental results demonstrate that our proposed approach outperforms the state-of-the-art and is flexible to generate various lengths of keyframe based summaries.

8.1.3 Sparse Dictionary Reconstruction based Video Summarization

The problem of video summarization can be reformulated as the problem of sparse dictionary reconstruction, where solutions can be derived with strong mathematical support. Chapter 5 and Chapter 6 present two algorithms to solve this problem. An L_0 constrained MSR model is firstly introduced, which reconstructs all the frames in a video as accurate as possible with as few frames as possible. Specifically, an off-line solution and an on-line solution are proposed by utilizing the proposed Minimum Sparse Reconstruction based principle. In addition, a POR criterion is proposed to intuitively guide users in selecting an appropriate length of the summary. Experimental

results demonstrate that our MSR based approaches outperform the state-of-the-art and are flexible to generate various lengths of keyframe based summaries. Moreover, our MSR based video summarization approaches can be utilized to summarize not only the structured videos, such as news, sports, or surveillance videos, but also the consumer videos without pre-defined structures. Then another $L_{2,0}$ constrained sparse dictionary selection model is proposed. An simultaneous orthogonal matching pursuit (SOMP) based keyframe extraction algorithm is devised to obtain an approximate solution for the proposed $L_{2,0}$ based problem directly without smoothing the penalty function. The experimental results are also very promising.

8.1.4 *StoryImaging* - A Text to Image Visualization System

Chapter 7 presents *StoryImaging*, a system that generates visual representation of a given textual story. This system contains streamlined components: text analysis, image search, search result re-ranking, near duplicate removal, and image clustering. Because the web search engine and images are utilized, any textual story can be illustrated visually by this system.

8.2 Future Work

8.2.1 Weighted Local Keypoint for Keyframe Selection

In our keypoint-based keyframe selection method (Chapter 3) and our top-down approach (Chapter 4), every keypoint is weighted equally when it contributes to keyframe selection. However, some keypoint may have higher importance and should contribute more to identify suitable keyframes, for example keypoints in face regions during change of facial expression, keypoints on more prominent foreground objects than those on other objects or those in the background, etc. This information has not been utilized

yet in our current framework, and we plan to incorporate this to get more robust and accurate video summaries.

8.2.2 Weighted Scene for Video Summarization

In our top-down approach (Chapter 4), the importance of each scene is treated equally, which may not be true in many videos. Some scenes should weight higher as they present more important information. This is not limited to visual information, but also audio and textual information. Identifying those scenes and selecting more keyframes for them reflects a more accurate representation of the oringinal story.

8.2.3 Feature-Rich Multimedia Summarization

The *StoryImaging* system is our first working system for multimedia summarization. It is simple but effective. In the future, we will extend the current framework to further integrate other modalities such as video and audio in augmenting textual information. Moreover, related audio and textual information can also be utilized to summarize video content. In [88], we have also made some progress on using visual features to identify the actions within images. Our ultimate goal is to eventually achieve multimedia story-telling.

Bibliography

[1] Youtube Statistics. http://www.youtube.com/yt/press/statistics.html, 2012.12.20. 1.1

[2] Wolfgang Effelsberg Rainer Lienhart, Silvia Pfeiffer. Video abstracting. *Communications of the ACM*, 40(12):54–62, December 1997. 1.1

[3] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1):37, 2007. 1.1, 2

[4] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 1.1, 3.1, 3.2.1, 4.1, 4.3, 4.4.4

[5] Yueting Zhuang, Yong Rui, T.S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *IEEE International Conference on Image Processing*, pages 866–870, 1998. 2, 2.1.1, 2.4, 3.4.1

[6] Daniel F Dementhon, Vikrant Kobla, and David Doermann. Video summarization by curve simplification. In *ACM international conference on Multimedia*, pages 211–218, 1998. 2, 2.1.2

[7] Jaime G. Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *ACM SIGIR Conference*

on Research and Development in Information Retrieval (SIGIR), pages 335–336, Melbourne, Australia, August 1998. 2, 2.1.1

[8] Arthur G. Money and Harry Agius. Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19:121–143, 2008. 2

[9] Sandra Eliza Fontes De Avila, Ana Paula Brandão Lopes, et al. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32:56–68, 2011. 2.1.1, 2.4, 4.4.1, 4.4.2

[10] Marco Furini, Filippo Geraci, Manuela Montangero, and Marco Pellegrini. STIMO: Still and moving video storyboard for the web scenario. *Multimedia Tools and Applications*, 46:47–69, 2010. 2.1.1, 2.4, 4.4.1

[11] Padmavathi Mundur, Yong Rao, and Yelena Yesha. Keyframe-based video summarization using Delaunay clustering. *International Journal on Digital Libraries*, 6(2):219–232, 2006. 2.1.1, 2.4, 4.4.1

[12] Dimitrios Besiris, A. Makedonas, George Economou, and Spiros Fotopoulos. Combining graph connectivity & dominant set clustering for video summarization. *Multimed Tools Application*, 44:161–186, 2009. 2.1.1, 2.4, 4.4.1

[13] Chong-Wah Ngo, Yu-Fei Ma, and Hong-Jiang Zhang. Video summarization and scene detection by graph modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 15:296–305, 2005. 2.1.1, 2.4

[14] Yihong Gong and Xin Liu. Video summarization using singular value decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 174–180, 2000. 2.1.1, 2.4

[15] Guido M. Schuster Li, Zhu and Aggelos K. Katsaggelos. MINMAX optimal video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, 15:1245–1256, 2005. 2.1.1, 2.4

[16] Yingbo Li, Bernard Merialdo, Mickael Rouvier, and Georges Linares. Static and dynamic video summaries. In *ACM International Conference on Multimedia (MM)*, pages 1573–1576, Scottsdale, AZ, USA, November 2011. 2.1.1, 2.4

[17] Costas Panagiotakis, Anastasios Doulamis, and Georgios Tziritas. Equivalent key frames selection based on Iso-content principles. *IEEE Transactions on Circuits and Systems for Video Technology*, 19:447–451, 2009. 2.1.2, 2.4, 3.4.1

[18] Yang Cong, Junsong Yuan, and Jiebo Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Transactions on Multimedia*, 14(1):66–75, February 2012. 2.1.2, 2.4, 4.4.1, 4.4.4

[19] Nuno Vasconcelos and Andrew Lippman. Bayesian modeling of video editing and structure: Semantic features for video summarization and browsing. In *Image Processing, International Conference on*, pages 153–157. IEEE, 1998. 2.2, 2.4

[20] Ahmet Ekin, A Murat Tekalp, and Rajiv Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12(7):796–807, 2003. 2.2, 2.4

[21] B. Chen, J. Wang, and J. Wang. A novel video summarization based on mining the story-structure and semantic relations among concept entities. *IEEE Transactions on Multimedia*, 11(2):295–312, 2009. 2.2, 2.4

[22] Richang Hong, Jinhui Tang, Hung-Khoon Tan, Chong-Wah Ngo, Shuicheng Yan, and Tat-Seng Chua. Beyond search: Event-driven summarization for web videos.

ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), 7(4):Article 35, 2011. 2.2

[23] Meng Wang, Richang Hong, Guangda Li, Zheng-Jun Zha, Shuicheng Yan, and Tat-Seng Chua. Event driven web video summarization by tag localization and key-shot identification. *IEEE Transactions on Multimedia*, 14(4):975–985, August 2012. 2.2

[24] Jiebo Luo, Christophe Papin, and Kathleen Costello. Towards extracting semantically meaningful key frames from personal video clips: from humans to computers. *IEEE Transactions on Circuits and Systems for Video Technology*, 19: 289–301, 2009. 2.2, 2.4

[25] Yu-Fei Ma, Xian-Sheng Hua, Lie Lu, and Hong-Jiang Zhang. A generic framework of user attention model and its application in video summarization. *IEEE Transactions on Multimedia*, 7:907 – 919, 2005. 2.2, 2.3

[26] Evangelopoulos Georgios, Konstantinos Rapantzikos, Alexandros Potamianos, Petros Maragos, A. Zlatintsi, and Yair Avrithis. Movie summarization based on audio-visual saliency detection. In *IEEE International Conference on Image Processing*, pages 2528–2531, 2008. 2.2, 2.3

[27] Yu Qiu, Genliang Guan, Zhiyong Wang, and Dagan Feng. Improving news video annotation with semantic context. In *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, pages 214–219. IEEE, 2010. 2.2

[28] Zhiyong Wang, Genliang Guan, Yu Qiu, Li Zhuo, and Dagan Feng. Semantic context based refinement for news video annotation. *Multimedia tools and applications*, 67(3):607–627, 2013. 2.2

[29] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2714–2721, June 2013. 2.2, 2.4

[30] Jia-Yu Pan, Hyungjeong Yang, and Christos Faloutsos. MMSS: Multi-modal story-oriented video summarization. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 491–494. IEEE, 2004. 2.3

[31] Pei Dong, Zhiyong Wang, Li Zhuo, and Dagan Feng. Video summarization with visual and semantic features. *Advances in Multimedia Information Processing-PCM 2010*, pages 203–214, 2010. 2.3

[32] Evangelopoulos Georgios, Athanasia Zlatintsi, Alexandros Potamianos, Petros Maragos, Konstantinos Rapantzikos, Georgios Skoumas, and Yannis Avrithis. Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. *IEEE Trans. on Multimedia*, 15(7):1553 – 1568, November 2013. 2.3

[33] Genliang Guan, Zhiyong Wang, Shiyang Lu, Jeremiah Da Deng, and David Dagan Feng. Keypoint based keyframe selection. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4):729–734, 2013. 3, 4.1, 4.3, 4.4.4, 5.3.1, 5.4.1, 5.4.1, 6.3, 6.4.1, 6.4.1

[34] David Feng, W.C. Siu, and Hong Jiang Zhang, editors. *Multimedia Information Retrieval and Management*. Springer, 2003. 3.1

[35] Costas Panagiotakis, Nikos Pelekis, Ioannis Kopanakis, Emmanuel Ramasso, and Yannis Theodoridis. Segmentation and sampling of moving object trajectories based on representativeness. *IEEE Transactions on Knowledge and Data Engineering*, 24(7):1328–1343, Jul 2012. 3.1

[36] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10 (27):1615–1630, 2005. 3.2.1, 4.3

[37] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981. 3.2.1

[38] Shiyang Lu, Zhiyong Wang, Meng Wang, M. Ott, and Dagan Feng. Adaptive reference frame selection for near-duplicate video shot detection. In *IEEE International Conference on Image Processing*, pages 2341–2344, 2010. 3.2.2

[39] Richard M Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 40:85–103, 1972. 3.3

[40] Savvas A. Chatzichristofis and Yiannis S. Boutalis. CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *International Conference on Computer Vision Systems*, pages 312–322, 2008. 3.4.1, 4.2, 7.3.5

[41] Chanop Silpa-Anan and Richard Hartley. Optimised kd-trees for fast image descriptor matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 3.4.4

[42] Genliang Guan, Zhiyong Wang, Shaohui Mei, Max Ott, Mingyi He, and David Dagan Feng. A top-down approach for video summarization. *ACM Transactions on Multimedia Computing, Communications and Applications*, 11(1):4:1–4:21, September 2014. 4

[43] Liangliang Cao, Yadong Mu, Apostol Natsev, Shih-Fu Chang, Gang Hua, and John R. Smith. Scene aligned pooling for complex video recognition. In *The*

European Conference on Computer Vision (ECCV), pages 688–701, Firenze, Italy, October 2012. 4.1

[44] Wan-Lei Zhao, Chong-Wah Ngo, Hung-Khoon Tan, and Xiao Wu. Near-duplicate keyframe identification with interest point matching and pattern learning. *IEEE Transactions on Multimedia*, 9:1037–1048, 2007. 4.1

[45] Yan-Tao Zheng, Shi-Yong Neo, Tat-Seng Chua, and Qi Tian. The use of temporal, semantic and visual partitioning model for efficient near-duplicate keyframe detection in large scale news corpus. In *ACM International Conference on Image and Video Retrieval*, pages 409–416, 2007. 4.1

[46] Gentao Liu, Xiangming Wen, Wei Zheng, and Peizhou He. Shot boundary detection and keyframe extraction based on scale invariant feature transform. In *IEEE/ACIS International Conference on Computer and Information Science*, pages 1126–1130, 2009. 4.1

[47] Jun Li, Youdong Ding, Yunyu Shi, and Wei Li. A divide-and-rule scheme for shot boundary detection based on sift. *International Journal of Digital Content Technology and its Applications*, 4:202–214, 2010. 4.1

[48] Maguelonne Heritier, Langis Gagnon, and Samuel Foucher. Places clustering of full-length film key-frames using latent aspect modeling over sift matches. *IEEE Transactions on Circuits and Systems for Video Technology*, 19:832–841, 2009. 4.1

[49] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *the Seventeenth International Conference on Machine Learning*, pages 727–734, 2000. 4.1, 4.2, 4.2, 4.2

[50] Genliang Guan, Zhiyong Wang, Kaimin Yu, Shaohui Mei, Mingyi He, and Dagan Feng. Video summarization with global and local features. In *IEEE International Conference on Multimedia and Expo Workshops*, pages 570–575, 2012. 4.1, 4.3, 4.4.4

[51] Ulrike Luxburg. A tutorial on spectral clustering. *Journal Statistics and Computing*, 17(4):395–416, December 2007. 4.2

[52] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, February 2007. 4.2

[53] Radhakrishna Achantay, Sheila Hemamiz, Francisco Estraday, and Sabine Susstrunky. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009. 4.3.1, 4.4.3

[54] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, 2009. 4.3.2

[55] Shaohui Mei, Genliang Guan, Zhiyong Wang, Shuai Wan, Mingyi He, and David Dagan Feng. Video summarization via minimum sparse reconstruction. *Pattern Recognition*, 48(2):522–533, 2015. 5

[56] Yang Cong, Junsong Yuan, and Jiebo Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Transactions on Multimedia*, 14(1):66–75, 2012. 5.1, 5.2.2, 5.3.2, 5.4.1, 5.4.1, 6.1, 6.3, 6.4.1, 6.4.1

[57] Jianxin Wu, Henrik I Christensen, and James M Rehg. Visual place categorization: Problem, dataset, and algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4763–4770. IEEE, 2009. 5.2.2

[58] Jianxin Wu and Jim M Rehg. Centrist: A visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8): 1489–1501, 2011. 5.2.2

[59] Sandra Eliza Fontes de Avila and Ana Paula Brandão Lopes and. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011. 5.4.1, 5.4.1, 5.4.2, 6.4.1, 6.4.1

[60] Open Video Project. http://www.open-video.org/, 2011.11.25. URL http://www.open-video.org/. 5.4.1, 6.4.1

[61] Padmavathi Mundur, Yong Rao, and Yelena Yesha. Keyframe-based video summarization using delaunay clustering. *International Journal on Digital Libraries*, 6(2):219–232, 2006. 5.4.1, 6.4.1

[62] Marco Furini, Filippo Geraci, Manuela Montangero, and Marco Pellegrini. Stimo: Still and moving video storyboard for the web scenario. *Multimedia Tools and Applications*, 46(1):47–69, 2010. 5.4.1, 6.4.1

[63] Da Deng. Content-based image collection summarization and comparison using self-organizing maps. *Pattern recognition*, 40(2):718–727, 2007. 5.5

[64] Chunlei Yang, Jialie Shen, Jinye Peng, and Jianping Fan. Image collection summarization via dictionary learning for sparse representation. *Pattern Recognition*, 46(3):948–961, 2013. 5.5

[65] Shaohui Mei, Genliang Guan, Zhiyong Wang, Mingyi He, Xian-Sheng Hua, and David Dagan Feng. L2,0 constrained sparse dictionary selection for video summarization. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2014. 6

[66] Anna C. Gilbert Tropp, Joel A. and Martin J. Strauss. Algorithms for simultaneous sparse approximation. part I: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006. 6.3

[67] Joel A Tropp. Algorithms for simultaneous sparse approximation. part II: Convex relaxation. *Signal Processing*, 86(3):589–602, 2006. 6.3

[68] Ramin Rezaiifar Pati, Yagyensh Chandra and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 40–44. IEEE, 1993. 6.3

[69] Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53 (12):4655–4666, 2007. 6.3

[70] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978. 6.3

[71] XXX. *StoryImaging*: A media-rich presentation system for textual stories. In *ACM International Conference on Multimedia*, Scottsdale, Arizona, USA, November 2011. 7

[72] Gary R. Bertoline. Visual science: An emerging discipline. *Journal for Geometry and Graphics*, 2:181–187, 1998. 7.1

[73] Zhihui Fang. Illustrations, text, and the child reader: What are pictures in children's storybooks for? *Reading Horizons*, 37:130, 1996. 7.1

[74] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999. 7.1

[75] Dhiraj Joshi, James Z. Wang, and Jia Li. The story picturing engine—a system for automatic text illustration. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2:68–89, 2006. 7.2

[76] Diogo Delgadoand, Joao Magalhaes, and Nuno Correia. Assisted news reading with automated illustration. In *MM '10 Proceedings of the international conference on Multimedia*, pages 1647–1650, 2010. 7.2

[77] Haojie Li, Jinhui Tang, Guangda Li, and Tat-Seng Chua. Word2image: towards visual interpreting of words. In *MM '08 Proceeding of the 16th ACM International Conference on Multimedia*, pages 813–816, 2008. 7.2

[78] Yansong Feng and Mirella Lapata. Topic models for image annotation and text illustration. In *The 2010 Annual Conference of the North American Chapter of the ACL (NAACL): Human Language Technologies (HLT)*, pages 831–839, Los Angeles, California, USA, June 2010. 7.2

[79] Meng Wang, Kuiyuan Yang, Xian-Sheng Hua, and Hong-Jiang Zhang. Visual tag dictionary: interpreting tags with visual words. In *Proceedings of the 1st Workshop on Web-scale Multimedia Corpus*, pages 1–8, 2009. 7.2

[80] Deng Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 7.2

[81] Bilyana Taneva, Mouna Kacimi, and Gerhard Weikum. Gathering and ranking photos of named entities with high precision, high recall, and diversity. In *ACM international conference on Web search and data mining (WSDM)*, pages 431–440, New York, USA, Feb 2010. 7.2

[82] Xin Lu, Yanwei Pang, Qiang Hao, and Lei Zhang. Visualizing textual travelogue with location-relevant images. In *International Workshop on Location Based Social Networks (LBSN '09)*, pages 65–68, Seattle, WA, USA, November 2009. 7.2

[83] Dipanjan Das and Andre FT Martins. A survey on automatic text summarization. Technical report, 2007. 7.3.1

[84] Jahna Otterbacher, Dragomir Radev, and Omer Kareem. Hierarchical summarization for delivering information to mobile devices. *Information Processing & Management*, 44:931–947, 2008. 7.3.1

[85] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pages 277–286, 1995. 7.3.4

[86] Reinier H. van Leuken, Lluis Garcia, Ximena Olivares, and Roelof van Zwol. Visual diversification of image search results. In *Proceedings of the 18th International Conference on World Wide Web*, pages 341–350, 2009. 7.3.5

[87] Marcie Begleiter. *From Word to Image: Storyboarding and the Filmmaking Process*. Michael Wiese Productions - 2nd edition, 2010. 7.5

[88] Gang Tian, Genliang Guan, Zhiyong Wang, and Dagan Feng. What is happening: annotating images with verbs. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1077–1080. ACM, 2012. 8.2.3