การตรวจหาเหตุการณ์ซ้ำซ้อนอัตโนมัติสำหรับวีดิทัศน์

นายณรงค์ศักดิ์ พุดเผือก

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

AUTOMATIC REDUNDANT EVENT DETECTION FOR VIDEO

Mr Narongsak Putpuek

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Computer Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic year 2011

Thesis Title	AUTOMATIC REDUNDANT EVENT DETECTION FOR VIDEO
By	Mr Narongsak Putpuek
Field of Study	Computer Science
Thesis Advisor	Assistant Professor Nagul Cooharojananone, Ph.D.
Thesis Co-advisor	Professor Chidchanok Lursinsap, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

……………………………………………….. Dean of the Faculty of Science
(Professor Supot Hannongbua, Dr.rer.nat.)

THESIS COMMITTEE

……………………………………………….. Chairman
(Assistant Professor Rajalida Lipikorn, Ph.D.)

……………………………………………….. Thesis Advisor
(Assistant Professor Nagul Cooharojananone, Ph.D.)

……………………………………………….. Thesis Co-advisor
(Professor Chidchanok Lursinsap, Ph.D.)

……………………………………………….. Examiner
(Suphakant Phimoltares, Ph.D.)

……………………………………………….. Examiner
(Supatana Auethavekiat, Ph.D.)

……………………………………………….. External Examiner
(Associate Professor Nopporn Chotikakamthorn, Ph.D.)

ณรงค์ศักดิ์ พุดเผือก : การตรวจหาเหตุการณ์ซ้ำซ้อนอัตโนมัติสำหรับวีดิทัศน์. (AUTOMATIC REDUNDANT EVENT DETECTION FOR VIDEO) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.นกุล คูหะโรจนานนท์, อ. ที่ปรึกษาวิทยานิพนธ์ร่วม ศ.ดร. ชิดชนก เหลือสินทรัพย์, 65 หน้า.

วิทยานิพนธ์ฉบับนี้เสนอวิธีการใหม่ซึ่งสามารถระบุตำแหน่งการถ่ายซ้ำสำหรับวีดีทัศน์ภาพยนต์ที่ยังไม่ได้ตัดต่อได้ โดยวิธีการนี้ขั้นแรกจะทำการแบ่งวีดีทัศน์ออกเป็นโครงสร้างพื้นฐาน เรียกว่าช๊อต โดยใช้วิธีการตรวจหาขอบเขตของช๊อตอัตโนมัติด้วยคุณลักษณะของเอสวีดีแบบเฉพาะส่วนและการจัดกลุ่มแบบเคมีน จากช๊อตที่ได้นั้นบางช๊อตจะประกอบไปด้วยเฟรมสีเดียว เฟรมทดสอบสี และเฟรมป้ายสเลท จะถูกขจัดออกไปด้วยขั้นตอนวิธีที่นำเสนอและวิธีการเอ็นดีเค จากนั้นช๊อตที่เหลือแต่ละเฟรมจะถูกสกัดคุณลักษณะแบบเฉพาะส่วนด้วยขั้นตอนวิธีซิบ แล้วทำการคำนวณหาความใกล้เคียงของเฟรมที่อยู่ต่อเนื่องกันโดยใช้วิธีการจับคู่ของซิบ และเปลี่ยนผลที่ได้ไปอยู่ในรูปของอัขระ ซึ่งอัขระที่ได้นั้นจะถูกนำมาต่อกันเป็นลำดับของอักขระเพื่อใช้เป็นตัวแทนของช๊อตนั้นๆ การหาความเหมือนของสองลับดับของอักขระจะใช้ขั้นตอนวิธีหาลำดับย่อยของอัขระเหมือนกันที่ยาวที่สุด ในการทดลอง ลำดับแรกทำการเปรียบผลลัพธ์จากวิธีการตรวจหาขอบเขตของช๊อตอัตโนมัติที่นำเสนอ กับวิธีการอื่น ลำดับที่สองทำการเปรียบเทียบผลลัพธ์ของการตรวจหาช๊อตการถ่ายซ้ำกับวิธีการอื่น ซึ่งจากผลการทดลองสนับสนุนว่าวิธีการที่นำเสนอมีความถูกต้องสูงอย่างมีนัยสำคัญ สำหรับการตรวจหาการถ่ายซ้ำในวีดีทัศน์ที่ยังไม่ได้ตัดต่อ

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต..
สาขาวิชา วิทยาการคอมพิวเตอร์.................. ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก..............
ปีการศึกษา 2554.................................... ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม...............

4973881523 : MAJOR COMPUTER SCIENCE

KEYWORDS: SEQUENCE MATCHING/ RETAKE DETECTION/ SIFT/ LCS/ RUSHES VIDEO/ SINGULAR VALUE DECOMPOSITION

NARONGSAK PUTPUEK : AUTOMATIC REDUNDANT EVENT DETECTION FOR VIDEO. ADVISOR : ASST. PROF. NAGUL COOHAROJANANONE, Ph.D., CO-ADVISOR: PROF. CHIDCHANOK LURSINSAP, Ph.D., 65 pp.

In this dissertation, a new methodology has been proposed to determine retake in rushes video. In this methodology, the video is divided into *shots* by the proposed automatic Shot Boundary Detection (SBD), which uses local *Singular Value Decomposition (SVD)* and *k-means clustering*. Shots that contain a single color, color bars or clapper boards will be eliminated by our proposed algorithm and *Near-Duplicated Keyframe (NDK)*. In the remaining shots, the local features of each frame are extracted using *Scale-Invariant Feature Transform (SIFT) algorithm*. The similarity between consecutive frames is calculated using a *SIFT matching* and then converted into a string. The given string is then concatenated into a string sequence to use as a shot representative. The similarity between two sequences is evaluated by the *Longest Common Subsequence algorithm (LCS)*. In the experiment, first, our automatic shot boundary detection is compared with conventional technique. Second, results of retake shots are compared with results from conventional technique. Results show that our proposed methodology provides a reasonably high degree of accuracy to detect a retake in rushes video.

Department : Mathematics and Computer Science	Student's Signature
Field of Study : Computer Science	Advisor's Signature
Academic Year : 2011	Co-advisor's Signature

Acknowledgements

I would like to express my sincere gratitude to my advisor Associate Professor Dr. Nagul Cooharojananone who has provided me with a great opportunity to do my research work under his guidance. His emphasis on suggestions and comments has motivated me during my Ph.D. study and research. I am especially deeply grateful to my co-advisor Professor Dr. Chidchanok Lursinsap who has provided me with great advice, guidance and encouragement throughout my Ph.D. study and research work.

I am grateful to Associate Professor Dr. Rajalida Lipikorn for her guidance and comments throughout my research work. Her advice helps me in the completion of my research work. I am grateful to Associate Professor Suchada Siripant for providing me with an excellent work environment in the Advanced Virtual and Intelligent Computing Center (AVIC) through all my Ph.D. study and research work.

I would like to thank the Rajabhat Rajanagarindra University for founding source. This found gave me a chance to pursue Ph.D. program. I would like to thank the National Institute of Informatics (NII) for founding source with one year of the internship program.

I am grateful to Professor Dr. Shin'ichi Satoh for provided me with guidance, a comment in the meeting and founding source for conference meeting in oversea. To the all members in Satoh's laboratory, sincerely thanks for your help and share their knowledge during my research in NII, Japan. I would like to thank all committee members for their comments and corrections of my dissertation at the committee meeting.

I would like to thank all members of AVIC's group for their helps, care and discussion on my Ph.D. study and research work. In addition, I would like to thank everyone who helped me directly or indirectly during my study at Chulalongkorn University. Finally, my special thanks to my family for their financial and moral support.

Contents

Page

Abstract (Thai) .... **iv**

Abstract (English) .... **v**

Acknowledgements .... **vi**

Contents .... **vii**

List of Figures .... **ix**

List of Tables .... **xi**

Chapter

I INTRODUCTION **1**
1.1 Introduction and Problem Review 1
1.2 Research Objective 3
1.3 Scopes of the Study 4
1.4 Contribution 4
1.5 Research Plans 4
1.6 Organization of the Dissertation 4

II BACKGROUND AND LITERATURES REVIEWS **5**
2.1 Background 5
2.1.1 TREC Video Retrieval Evaluation (TRECVID) 5
2.1.2 Video Structure 6
2.1.3 Rushes Video 7
2.1.4 Singular Value Decomposition (SVD) 8
2.1.5 k-Means Clustering 9
2.1.6 Longest Common Subsequence 10
2.1.7 Scale Invariant Feature Transform (SIFT) 11
2.1.8 SIFT Features Matching 12
2.1.9 Near Duplicate Keyframe (NDK) 13
2.2 Literature Review 14
2.2.1 Video Shot Boundary Detection 14
2.2.2 Retake Detection 15

III PROPOSED METHOD **18**
3.1 Proposed Framework 18
3.2 Video Representation 18
3.3 Shot Boundary Detection 19
3.3.1 Local Feature Extraction 19
3.3.2 Feature Similarity Measure 20
3.3.3 Shot Clustering 21

Chapter Page

3.4 Junk Elimination 22
3.4.1 Color Bars 24
3.4.2 Single Color frames 25
3.4.3 Clapper Boards 26
3.5 Feature Extraction 27
3.6 Retake Detection 33

IV EXPERIMENTAL RESULTS 36
4.1 Data sets 36
4.1.1 Shot Boundary Detection 36
4.1.2 Junk Elimination, Feature Extraction and Retake Detection 36
4.2 Performance Evaluation 37
4.3 Shot Boundary Detection 38
4.3.1 Comparison for number of block selection 38
4.3.2 Comparison for number of large value removal 39
4.3.3 Comparison for performance 41
4.4 Junk Elimination 43
4.4.1 Color Bars Removal 46
4.4.2 Single Color Removal 47
4.4.3 Clapper Board Removal 49
4.5 Retake Detection 50
4.5.1 Manually extract the shot boundary 50
4.5.2 Automatic extract the shot boundary 51

V CONCLUSION 54
5.1 Conclusion 54
5.2 Future Work 54

REFERENCES 55

Appendix 63

Biography 65

List of Figures

Figure Page

1.1 Example of retake from video MRS025913. 1
2.1 The structure of video. 7
2.2 The useless contents are included in rushes video. (a) The color bars. (b) The Single color frame. (c) The clapper board. 7
2.3 The repetition shots with the same or near-identical settings are taken from video MRS035123. 8
2.4 An example of Singular Value Decomposition (SVD). 9
2.5 An example of Singular Value. (a) An example of image. (b) The first ten singular values are determined from the given image. 9
2.6 An example result of k-means method. (a) The input patterns. (b) The input patterns are partitioned into two clusters with $k = 2$. 10
2.7 An example of detected SIFT features along with their locations and scales. 12
2.8 An example of matched SIFT features between the two frames. 13
3.1 Proposed framework. 18
3.2 An example of object appears in the video sequence. 19
3.3 An example of frame is divided into 4 ×4 blocks with the layout of local feature. 20
3.4 Example of the similarity values are computed between frame f_t and f_{t+1} with 4×4 blocks. 21
3.5 Example of two classes boundary (Normal boundary and Cut boundary). 22
3.6 Plot of SVD distance between frame pair. 23
3.7 Example of normalized SVD. 23
3.8 Example of clustering result. 24
3.9 Example of color bars are included in rushes video. 24
3.10 Example of single color are included in rushes video. 25
3.11 Example of clapper boards are included in rushes video. 26
3.12 Examples of clapper boards appear in many situation such as scale, rotation and illumination changes. 27
3.13 A frame is divided into 5 × 5 blocks. 28
3.14 An example of SIFT matching result is matched between frames f_t and f_{t+1} at block1. . . 30
3.15 An example of SIFT matching result. 30
3.16 An example of blocks selection.. 31
3.17 A frame is divided into 5 × 5 blocks and the set of English alphabet are assigned. 31
3.18 An example of five frames that are encoded into a string sequence. 32
3.19 An example of takes that appear in the two scenes. 33
3.20 An example of common subsequence is determined from shot9 and shot10. 35
4.1 Clustering results of video BG_2408. The left side of vertical line is a normal boundary, and the right side of vertical line is a cut boundary. 39
4.2 Examples of missed detections by the proposed method are taken from video BG_35187. . 44
4.3 Examples of missed and false detection by Le[63] are taken from video BG_35187. 45
4.4 Examples of missed and false detections by the proposed method are taken from video BG_37879. 45
4.5 Examples of missed and false detections by Le[63] are taken from video BG_37879. 46
4.6 Examples of false detections by the proposed method are taken from video MRS035126 and MRS0145343. 48

Figure Page

4.7 Examples of missed detections by the NDK method are taken from video MRS145343. . . 50
4.8 Examples of retake that is taken from video MRS151585. 52
4.9 Examples of retake that is taken from video MRS150072. 52

List of Tables

Table Page

4.1 TRECVID shot Boundary detection data set. 36
4.2 Experimental rushes video for tested and evaluated. 37
4.3 Performance comparison of number of blocks selection. 40
4.4 Performance comparison for the parameter θ is varied between 0 to 63. 42
4.5 Performance comparison of proposed method and existing method. 44
4.6 The result of color bars removal 47
4.7 The result of single color removal 48
4.8 The result of clapper boards removal 49
4.9 Performance comparison with manual shot boundary detection. 51
4.10 Performance comparison with automatic shot boundary detection. 53

CHAPTER I

INTRODUCTION

1.1 Introduction and Problem Review

Nowadays, as the advancement in digital technology has grown significantly such as data acquisition, storage and communication technologies. Then, it produces the availability of video data which increases at an exponential rate. In the post-production management of digital cinema contents, a large amount of the raw video (rushes video) needs to viewed and organized due to the importance of the content that is required in the final version. In general, rushes videos are the raw videos which are recorded from filmmaking production. A basic structure of rushes video is called a scene which recording followed in the script. A scene is typically recorded many times according to the requirement of director, with various setting and unexpected mistakes [1, 2]. Therefore, there are two types of content in the rushes video: useless content and redundant content. The useless content is the shots that are not relevant to the main content of the video such as color bars, single color frames and clapper boards. The redundant content is called a **retake**, a repetition shots with the same or near-identical setting. Figure 1.1 shown an example of the retake which consists of two takes from video MRS025913. Then, a retake is needed to detect, and then the best take should be selected before the post-production. However, It is time consuming to manually eliminate retakes from long vide sequences, therefore, a method for automatically but reliably eliminating retakes would be useful.

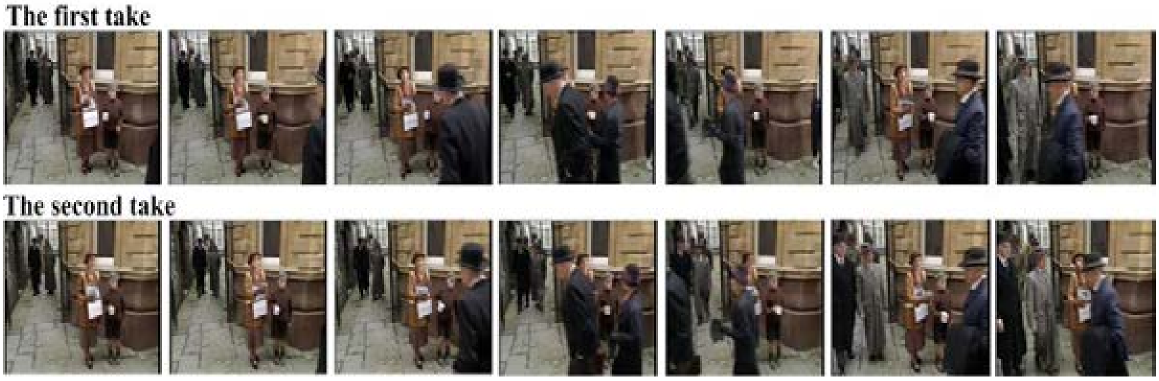


Figure 1.1 Example of retake from video MRS025913.

Several approaches have been proposed for the redundant detection of the video. [3] proposes a framework for detecting a redundant video. It used a sequence shape similarity metric and k-means clustering algorithm. Linjin [4] focuses on detecting and removes the redundant video content by using the hierarchical agglomerative cluster (HAC) and the Smith-Waterman algorithm to produce the video summary. In sports videos , a method [5] proposes to analyze and to detect the redundant data by using the characteristics of main colors and multi-region segmentation algorithm. [6, 7] focus on matching commercial film clips by using binary signature and simple distance matching method. However, conventional redundant detection methods cannot provide the efficiency results for rushes video. Due to some significant differences from another video. Moreover, these videos are unedited and contain useless and redundant content.

Recently, there are several researches proposed the method to detect the retake in rushes videos [8–17]. As for the first step of automatic video processing systems, the continuous video sequences are usually segment into shots that are the basic video units. The automatic shot boundary detection is applied by using pixel different with threshold [15–17], histogram different with threshold [8, 9, 13] or adaptive threshold [10] and color-texture with the analysis of temporal slices [11]. Keyframe selection methods are employed to extract the shots or sub-shots representative [8–17]. In order to detect the retake in rushes video, the [8] uses of hierarchical agglomerative clustering algorithm was based on sub-shots that are represented by its average on local histogram. Liu et al. [14] proposed a multi-state clustering algorithm based on keyframes of sub-shot. In [11] detects the retake by using keyframe and speech transcript comparison based on directed graph. However, the previous methods are limited because a short or sub-shot representative is depended on keyframe selection methods, and moreover the lowest number of keyframes cannot provide high efficiency for clustering method.

Another approach has been proposed the method to detect retake based on sequence matching [18–20]. In [18, 19], the rushes video is decomposed into one-second segments, and then these segments are clustered by using a hierarchical classification to group into long segments. The given long segments are used to construct the alignment matrix by using frame-based and the Smith-Waterman algorithm. The retakes are detected by searching for white rectangle areas in the alignment matrix. However, this method is limited because the alignment matrix is constructed by using frame-based which is computationally expensive, and moreover it requires manually adjusted threshold for retake detection. A distance measure approach based on the Longest Common Subsequence (LCS) algorithm is presented in [20], rushes video is decomposed into segments using shot boundary detection based on SVM classifier. The similarities for all shots are computed based on LCS algorithm, and then two shots will be merged if the value of shot similarity larger than the predefined threshold.

Single linkage clustering is used to determine the retakes based on the value of shot similarities. The result shows that this method has a good performance for detecting the retake of the rushes video. However, it cannot differentiate between the scenes with little action and shot in the same room. Due to the difference in visual information and activity between the scenes is very small.

In this dissertation proposes a method to detect the retake of rushes video by using the characteristic of a video sequence, the object recognition method was based on Scale-invariant feature transform (SIFT), the characteristic of retake and the Longest Common Subsequence (LCS) algorithm. This method uses the characteristic of a video sequence that can be represented as the sequence of objects. In rushes video, the retake is a repetition shots with the same or near-identical setting, this then it will be has a common subsequence of the object pattern. To detect a common subsequence, the sequence of object location is encoded into a sequence of string pattern by using the object recognition method based on SIFT feature and the grid method. Then, LCS algorithm can be used to find a common subsequence, an indeed was recently reported to provide the best performance when using for matching the sequence [21]. From the characteristic of retake, each retake has two or more takes that appear as an order of sequence. Taking this characteristic into account, the method can detect the retake by finding a common subsequence of string pattern based on a simple algorithm and LCS algorithm. The framework of this proposed method has been designed into four steps. Step 1, the rushes video is divided into structure call shots using our proposed automatic Shot Boundary Detection (SBD) based on Singular Value Decomposition (SVD) and k-means clustering. Shots that contain a single color, color bars or clapper boards will be eliminated by our proposed algorithm using Near Duplicated Keyframe (NDK) in step 2. Step 3, in the remaining shots, the local feature of each frame is extracted using SIFT algorithm. The similarity between consecutive frames is calculated using a SIFT matching and then converted into a string. The given string is then concatenated into a string sequence to use as shot representative. In step four, the similarity between two sequences is evaluated by the LCS algorithm. The simple algorithm is performed to detect the retake by using its characteristic.

1.2 Research Objective

The main objectives of this dissertation are the following:

(a) To develop a new technique for retake detection of rushes videos, which enhances the performance in terms of recall and precision.

(b) To develop a new technique for abrupt shot boundary detection, which enhances the performance in terms of recall and precision.

1.3 Scopes of the Study

In this dissertation, the scope of work is constrained as follows:

1. For retake detection, the experimental results are based on TRECVID 2007-2008 BBC rushes summarization data sets.
2. For video shot boundary detection, The proposed method is focused only on detection abrupt cut shot boundary. The experimental results are based on TRECVID 2004 and 2007 video shot boundary detection data sets.

1.4 Contribution

This dissertation proposed a new method to detect the retake of rushes video based on Scale-Invariant Feature Transform (SIFT) and Longest Common Subsequence (LCS). This method uses the characteristics of retake such that the retake has a common subsequence of the object pattern. The framework of this proposed method is designed into four steps. Firstly, the rushes video is divided into shots and, then, shots containing a single color, color bars or clapper boards are eliminated. In the remaining shots, the local features of each frame are extracted using SIFT algorithm. The similarity between consecutive frames is calculated by using a SIFT matching and, then, converted into a string. The given string is, then, concatenated into a string sequence and the LCS algorithm evaluates the similarity between two sequences. The simple algorithm is performed to detect the retake by using its characteristic. This proposed method was tested and evaluated with other existing techniques based on the available benchmark data sets.

1.5 Research Plans

1. Study and review the related papers that are related in retake detection.
2. Develop a new technique for abrupt cut shot boundary detection and retake detection.
3. Develop and test with benchmark data set.
4. Compare the results with the other techniques.
5. Analyze the experimental results and summarize the outcomes.

1.6 Organization of the Dissertation

The rest of this dissertation is organized into five chapters. Chapter II reviews the background information and the methods related to the proposed method. Chapter III describes the new technique for abrupt cut shot boundary detection and retake detection. Chapter IV the experimental results being presented. Chapter V the conclusion and future work are presented.

CHAPTER II

BACKGROUND AND LITERATURES REVIEWS

In this chapter, the theoretical background on Video structure, Rushes video, Singular Value Decomposition (SVD), K-means clustering, Longest Common Subsequence, Scale Invariant Feature Transform (SIFT), SIFT Features Matching, Near Duplicate Keyframe (NDK) are described. The previous work on video shot boundary detection and retake detection that related to the proposed method are also reviewed and discussed.

2.1 Background

In this section, TRECVID, Video structure, Rushes video, Singular Value Decomposition (SVD), K-means clustering, Longest Common Subsequence, Scale Invariant Feature Transform (SIFT), SIFT Features Matching, Near Duplicate Keyframe (NDK) are reviewed.

2.1.1 TREC Video Retrieval Evaluation (TRECVID)

The Text Retrieval Conference's (TREC's) Video Retrieval Evaluation (TRECVID) was a TRECstyle video analysis and retrieval evaluation, the goal of which remains to promote progress in contentbased exploitation of digital video via open, metricsbased evaluation [22–30]. TRECVID is funded by the National Institute of Standards and Technology (NIST) and other US government agencies. Many organizations and individuals worldwide also contributed significant time and effort. From 2001 to 2011, there are consist of tasks as following:

1. Shot Boundary Detection
2. Known-item(s) Search
3. General Statements of Information Need
4. Feature Extraction
5. Search
6. Story Segmentation

7. Low-level Feature Extraction (Camera Motion)
8. High-Level Feature Extraction
9. Explore BCC Rushes
10. Rushes Summarization
11. Surveillance Event Detection (SED)
12. Content-based Copy Detection (CCD)
13. Semantic Indexing (SIN)
14. Known-item Search (KIS)
15. Instance Search (INS)
16. Event Detection in Internet Multimedia (MED)

2.1.2 Video Structure

In recent times, videos are widely used in many research fields such as video indexing [31], video classification [10], commercial film classification [32] and video summarization [33]. In order to analyzing content, videos are needed to be divide into subunits. Generally, a video can be organized into a syntactic structure based on the video production [34]. This structure consists of frames, shots and scenes [33, 35–37].

- Frames are the basic component in video which is represented by a static image.
- A shot contains with the contiguous sequence of frames which are defined a boundary by using a transition between the image content.
- A scene is the combination of shots that represent a different camera shot with the same content.

The video structure is schematically shown in Figure 2.1.

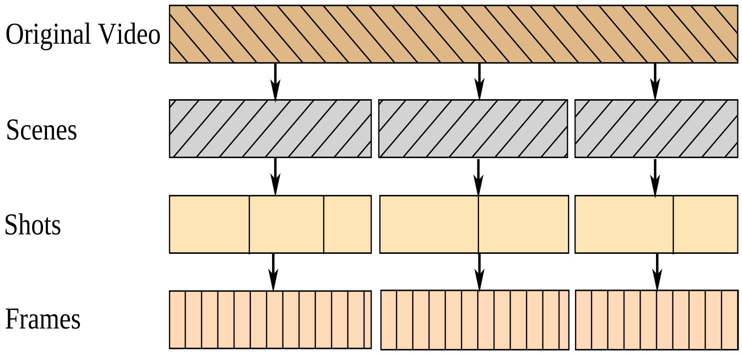
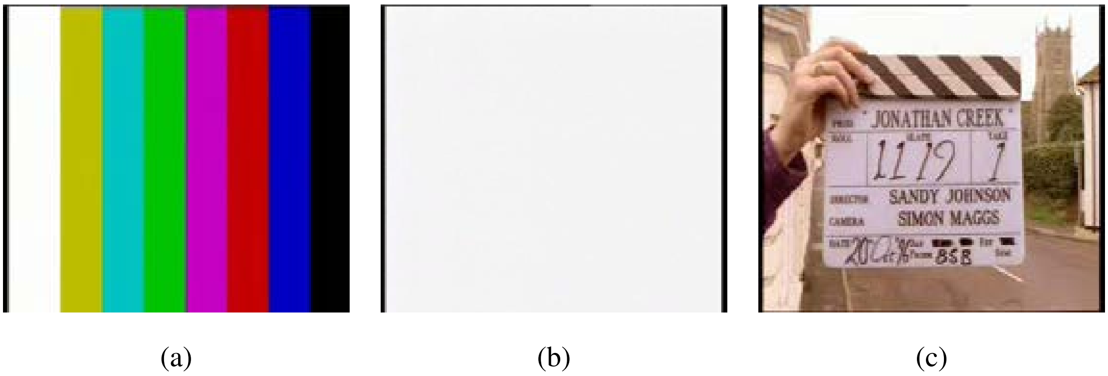


Figure 2.1 The structure of video.

2.1.3 Rushes Video

In general, the result from filmmaking is the rushes videos which are the raw recording from a digital video camera. They are obtained from the arrangement of movie production. During filmmaking process, a scene is typically taken several times due to some unexpected mistakes with an actor or the directors needs to ask for different performance. Therefore, the rushes video contains useless content and redundant content. The useless content is the shots that are not relevant to the main content of the video such as color bars, single color frames and clapper boards (see Figure 2.2). The redundant content is the repetition shots with the same or near-identical setting as shown in Figure 2.3.



(a) (b) (c)

Figure 2.2: The useless contents are included in rushes video. (a) The color bars. (b) The Single color frame. (c) The clapper board.

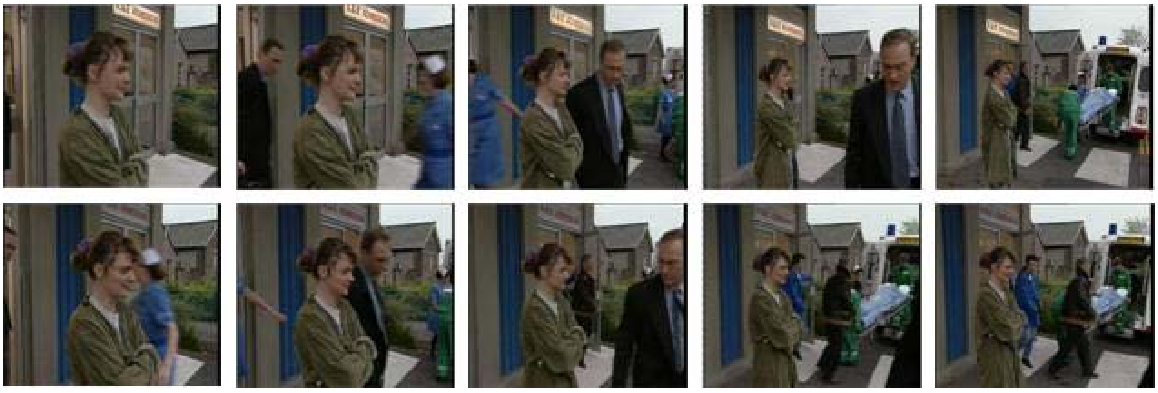


Figure 2.3: The repetition shots with the same or near-identical settings are taken from video MRS035123.

2.1.4 Singular Value Decomposition (SVD)

The matrix decomposition is one of the most gainful ideas in the theory of matrices. The theoretical utility of matrix decompositions has long been appreciated. More recently, they have become the mainstay of numerical linear algebra, where they serve as computational platforms from which a variety of problems can be solved. The singular value decomposition (SVD), one of the most useful decomposition of linear algebra, is a factorization and approximation theory which effectively reduces any matrix into a smaller invertible and square matrix [38–40].

In order to determine the singular value decomposition from a given rectangular matrix, Let A be a real $m \times n$ matrix with $m \geq n$. It can be factored in to the form

$$A = USV^T \tag{2.1}$$

where U and V are unitary matrices and S is a rectangular diagonal matrix of the same size as A. These diagonal elements are called the singular values of A. Then, we can assume that

$$S = diag(\sigma_1, \ldots, \sigma_n) \tag{2.2}$$

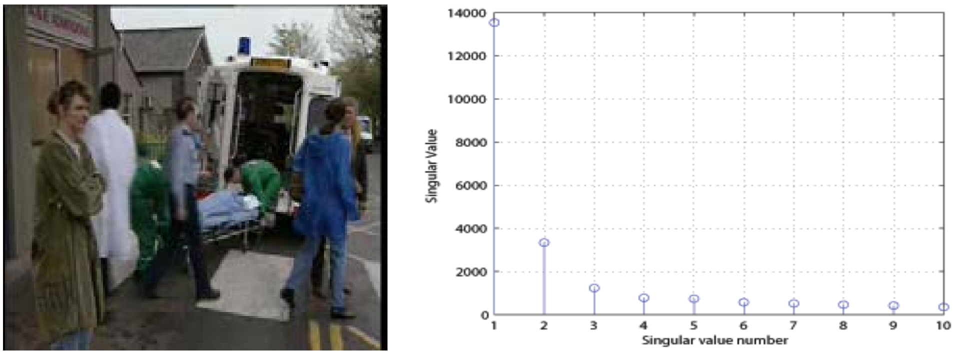
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0 \tag{2.3}$$

Thus, it can be proved that $rank(A) = r, \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_n = 0$. The example of SVD is shown in Figure 2.4.

$$\underset{A}{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}} = \underset{U}{\begin{bmatrix} -0.85 & -0.53 \\ -0.53 & -0.85 \end{bmatrix}} \underset{S}{\begin{bmatrix} 1.62 & 0 \\ 0 & 0.62 \end{bmatrix}} \underset{V}{\begin{bmatrix} -0.85 & 0.53 \\ -0.53 & -0.85 \end{bmatrix}}$$

Figure 2.4 An example of Singular Value Decomposition (SVD).

In digital image processing, several studies have already focused on the use of SVD for Image enhancement [41], video sequence matching [42] and video shot boundary detection [43]. By considering the input image as matrix A, and can be decomposes in to a singular value with the corresponding singular vector factorization. These values are useful information for discriminating image patterns or contents. Fig 2.5 is shown an example image and their singular values.



(a) (b)

Figure 2.5: An example of Singular Value. (a) An example of image. (b) The first ten singular values are determined from the given image.

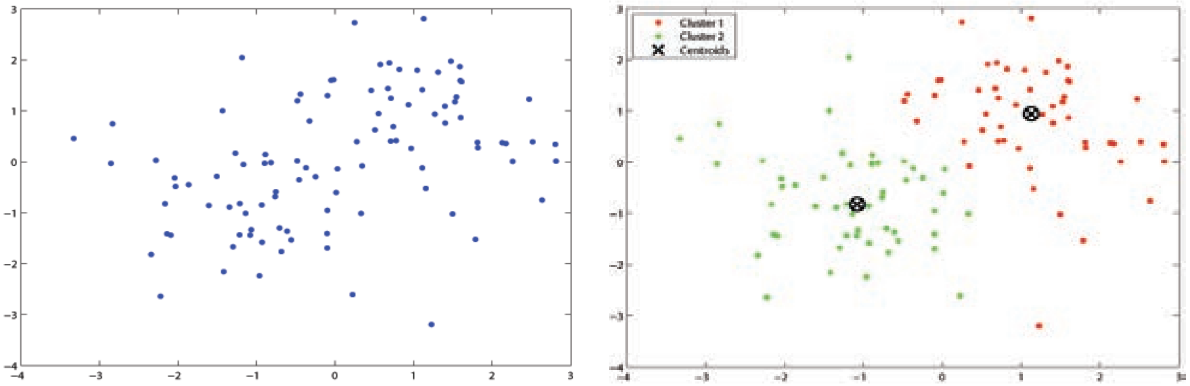
2.1.5 k-Means Clustering

The k-means method is the most commonly used algorithms for geometric clustering. This algorithm was originally proposed by Forgy [44], McQueen [45], and it is well known as Lloyd' algorithm [46]. The algorithm uses a local search method to partition n data points into k clusters. It starts with a random initial partition and keeps reassigning the n data points to k clusters based on the similarity between the n data points and the cluster centers until a convergence criterion is met.

Nowadays, the k-means method is still very popular and it has been applied in a widely areas in digital image processing and computer vision. Due to it's simplicity and its time complexity is $O(n)$, where n is the number of data points [47]. Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of patterns. Let $C = \{c_1, c_2, \ldots, c_k\}$ be a set of cluster centroids. The k-means algorithm has the following steps:

1. Arbitrarily choose k initial centers $c_1, c_2 \ldots, c_k$.
2. For each $i \in \{1, \ldots, k\}$, set the cluster C_i to be the set of patterns in P that are closer to c_i than c_j for all $j \neq i$.
3. For each $i \in \{1, \ldots, k\}$, set the cluster c_i to be the center of mass of all patterns in $C_i : c_i = \frac{1}{|C_i|} \sum_{p_j \in C_i} p_j$.
4. Repeat steps 1 and 2 until c_i and C_i no longer change. The partition of P is the set of clusters $C_1, C_2, \ldots, C_k$.

The example result of k-means method is shown in Fig 2.6. The random patterns are shown in Fig 2.6(a). The result for k-means algorithm with $k = 2$ is shown in Fig 2.6(b).



(a)

(b)

Figure 2.6: An example result of k-means method. (a) The input patterns. (b) The input patterns are partitioned into two clusters with $k = 2$.

2.1.6 Longest Common Subsequence

The LCS can be used to measure the similarity between two string sequences, and indeed was recently reported to provide the best performance when using distance measurements for clustering multiple takes of the same scene [21]. This is achieved since LCS provides a similarity of two or more string sequences by determining the length of each common subsequence. It implies that a sequence $C = c_1, c_2, ..., c_m$ is a subsequence of $A = a_1, a_2, ..., a_n$ if there exist indices $1 \leq i_1 < \cdots < i_m \leq n$ such that $C = a_{i1}, a_{i2}, ..., a_{im}$. We say that a sequence C is a common subsequence of a given two sequence A and B if C is a subsequence of both A and B. The sequence C can be computed by using a dynamic programming approach as following recurrence formula:

$$C[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \text{ ,} \\ C[i-1, j-1] + 1 & \text{if } i, j > 0 \text{ and } a_i = b_j \text{ ,} \\ max(C[i, j-1], C[i-1, j]) & \text{if } i, j > 0 \text{ and } a_i \neq b_j \text{ .} \end{cases} \tag{2.4}$$

where $C[i, j]$ is the length of a common subsequence.

2.1.7 Scale Invariant Feature Transform (SIFT)

Recently, the Scale Invariant Feature Transform (SIFT) algorithm presented [48–51]. SIFT contains a histogram representing the gradient orientation and magnitude information within a patch of images. The advantage of this feature is that it is invariant to image scaling, translation and rotation, and is also partially invariant to the illumination changes and affine or 3D projection. The SIFT feature can be applied to object recognition [52], image similarity [53] and near duplicate image identification [54]. The feature extraction is performed through the four successive steps as follows:

- Scale-space local extrema detection: The feature is detected by finding the locations that represent the maxima or minima difference-of-Gaussian function in scale space.
- Keypoint localization: The given keypoints from the previous step need to be extracted for localized information. Keypoints with a low contrast or poorly localized along an edge are rejected. The localized data is accomplished by fitting a 3D quadratic function to the local sample point. The quadratic function is computed using a Taylor expansion of the scale-space function.
- Orientation assignment: When each keypoint location is determined, an orientation must be assigned to it. The gradient magnitude and orientation are computed from each pixel of the region around the keypoint location, as per the following equation:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{2.5}$$

where $L(x, y)$ is an image sample, $m(x, y)$ is the gradient magnitude,

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1))/(L(x+1, y) - L(x-1, y))) \tag{2.6}$$

where $\theta(x, y)$ is an orientation.

- Keypoint descriptor: When each keypoint is assigned a scale and an orientation, the similarity - invariant patches are extracted. The region around each keypoint is divided into 16 (4×4) subregions. The gradient samples are accumulated into orientation histograms that epitomize the given subregions. Then, for each 16 subregions, an eight-orientation bin histogram is established and a 128 dimensional vector is constructed. In order to make it invariant to illumination change, histogram values with large gradients are reduced to a threshold and the feature vectors are normalized into unit lengths.

An example of SIFT features are shown in Figure 2.7.

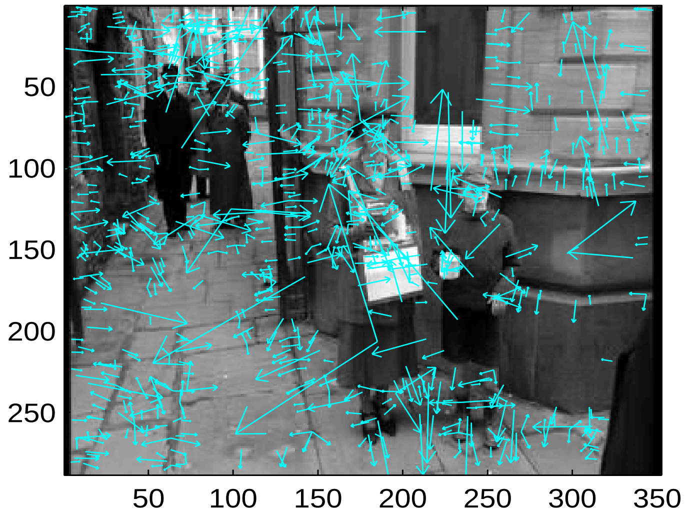


Figure 2.7 An example of detected SIFT features along with their locations and scales.

2.1.8 SIFT Features Matching

In order to match the keypoints from two images, Fan et al. [55] introduced a simple matching method based on the distinction of the keypoints. Let Kp_x being a keypoint from image X and Kp_y from image Y. Kp_x considered a match to Kp_y if Kp_x is the nearest neighbor of Kp_y in the descriptor's feature space. The nearest neighbor (NN) algorithm is used as the matching algorithm. It can be computed as the following equation:

$$\frac{D(Kp_y, Kp_x)^2}{D(Kp_y, \acute{K}p_x)^2} < \tau^2 \tag{2.7}$$

where $D(.,.)$ is the Euclidean distance between the descriptors of the two keypoints, $\acute{K}p_x$ is the second nearest keypoint of Kp_y in image X, and τ is a threshold for determining whether Kp_x and Kp_y are matched or not. An example of SIFT features matching is shown in Figure 2.8.

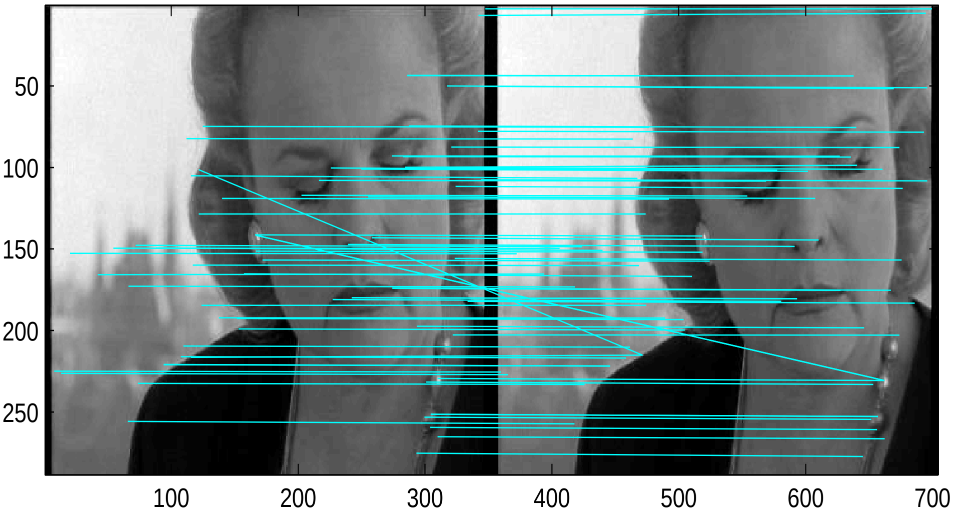


Figure 2.8 An example of matched SIFT features between the two frames.

2.1.9 Near Duplicate Keyframe (NDK)

Near-duplicate keyframes (NKDs) are the collection of keyframes which are similar or nearly duplicate of each other. It appears in the different situation due to a variety of capturing, in digitalization and editing conditions. The methodology used for identifying a pair of near-duplicate keyframes (NDK) is very useful for a variety of applications such as commercial film detection, news story classification and content-based video search. Ngo et al. [56, 57] introduced an efficient method to identify a near-duplicate keyframes in a broadcast domain based on interest point matching and pattern learning. This method is invariant to image scaling, translation, rotation, illumination changes, and affine or 3D projection. First, all keyframes extract the key points and match them by using the one-to-one symmetric (OOS) mapping strategy with an index structure LIP-IS. Finally, the degree of matching coherency in space are measured by evaluating the entropy of the patterns. They evaluate their method by using TRECVID-2004 broadcast videos. The results show that their method has a good performance in terms of recall and precision within a large margin. Therefore, this method can be used to eliminate the clapper boards s well as giving some detail in chapter three.

2.2 Literature Review

This section gives a review in shot boundary detection and retake detection that will be compared with the proposed method.

2.2.1 Video Shot Boundary Detection

Nowadays, the advancement in digital technology has grown significantly such as data acquisition, storage and communication technologies. It also produces the availability of video which increases at an exponential rate. Then, the techniques for browsing, retrieval, classification and summarization are needed. Therefore, the video data need to be organized into compact forms for extracting semantically meaningful information. Shot boundary detection is the most basic technique that is widely used as the first step to organize the video data into segments.

Based on the video production process, a shot is contained within a contiguous sequence of frames which defines a boundary by using a transition between the image content. According to TRECVID's categorization [58], there are mainly two types of transitions in a video sequence: The cut or abrupt shot boundary, as well as the gradual shot boundary. A cut shot boundary is a shot change that may occur between the two frames. A gradual shot boundary is a slow change that occurs over multiple frames.

As previously mentioned, in this study we focus on developing an efficient method to eliminate a retake in rushes video. According to the definition of rushes video [25, 26] , all rushes are unedited; therefore, they must consist of hard cut only. Then, the method for detecting a cut shot boundaries are performable for these video.

There are many approaches that have been proposed for shot boundary detection. The simplest approach is to determine the difference between the consecutive frames based on the global features, such as color histograms or pixel intensity [59]. Another method used the local features by dividing each input frame into blocks [8]. A threshold based is used to classify the hard cuts. The results showed that these methods are fast and simple. In [59], a comparison of hard cut detection method is presented. However, a threshold based method cannot differentiate between the hard cut and the large objects motion. Moreover, it is difficult to achieve equality efficiency for new video data. This problem can be overcome by using the clustering method. Suzuki et al. [60] proposed a method based on integrating multiple features and statistical pattern recognition. The method provides a good performance for hard cut detection. However, computation time is required due to integrating multiple features, especially motion vector and MDH. Furthermore, there is no clarity in the detail for statistical pattern recognition that they are used. Cernekova et al. [61, 62] proposed a technique based on the Singular Value Decomposition (SVD) and unsupervised clustering. SVD is applied on color histogram, and then classify a shot boundary by using a static threshold and a hypothesis test between two consecutive frames. Using the technique based on SVD, a dimension of features can be reduce. The experiments were conducted on the TRECVID 2004 test set and their method has a good performance in terms of recall and precision. However, this method is limited because it uses brute force matching to determine the number of clusters that is computationally expensive. Le et al. [63] presents an approach for video shot boundary detection based on visual information and Support Vector Machine (SVM). A training set performs by manually labeling all video frames into six classes that correspond to the transition types. The experiments were conducted on the TRECVID 2003 test set have shown that their approach is effective. However, this approach is limited due to high computation time and large memory are required with SVM for training and testing.

2.2.2 Retake Detection

Redundancy elimination for rushes video is a challenging task due to difficult repetitive segments, which are taken from the same scene, usually have different lengths and motion patterns. Recent work on retake detection [8, 10, 64–66] has proposed a technique using keyframes based as video shot that represent the clustering algorithm to eliminate redundancy. In [10] present a method to detect retake based on the tight clustering is produced via SIFT matching. Rushes video is segmented into shots using a simple method of applying an adaptive-threshold on the discontinuity. The first frame, the frame that has visual appearance's significantly different from the last frame and the last frames are selected as keyframes for each shot. Shots that are relevant to the useless content are filtered out. Then, shots are clustered via SIFT matching for redundancy elimination. Le [64] et al. introduced a method on how to detect retake based on clustering via color distributions. First, the input video is decomposed into fragments by comparing consecutive frames. These fragments are grouped by a clustering method and then, consecutive fragments are grouped into segments. Finally, the adjacent segments are merged if the distance between them falls below a threshold. Keyframes based and single-linkage clustering algorithm for retake detection is proposed in [8]. Rushes video is decomposed into shots and sub-shots using color histogram distance. The useless shots are then removed via color histogram detection. Then, a hierarchical agglomerative clustering algorithm based on average local histogram is used to eliminate retakes. In [65] proposed an algorithm to determine retakes based on hierarchical modeling of adaptive clustering. First, rush videos are hierarchically modeled using the formal language technique, then shot boundaries are applied to extract them and then construct a structuring hierarchical model of videos based on the concept of V-unit. The junk frames within this model are eliminated. Adaptive clustering is employed to group shots into clusters to determine and eliminate retakes. Noguchi et al. [66] introduced a method based on shots and k-means clustering. First, an input video is divided into shots by using difference measure between consecutive frames. Then, these shots are grouped by the k-means method, using color, motion and faces as features to detect and eliminate retakes. In [11] proposed a method to eliminate retake based on keyframes and directed graph search. An input video is decomposed into shots using the analysis of spatio-temporal slices extracted from the compressed domain. Keyframes are selected using detected high curvature points within each shot and junk shots are then filtered out. Shots are partitioned into sub-shots corresponding to different phases during video capture. The similarity between the two sub-shots are determined based on keyframe and speech transcript comparison. Then, a directed graph is constructed based on sub-shots similarity. The repetitive sub-shot detection is solved by searching for maximum complete subgraph.

Other approach [18–21] presents a technique based on sequence matching and clustering algorithm to detect and eliminate retakes that are related to the proposed method. Emilie et al. [18,19] proposed a method based on the detection of repetitive sequences, using a variant of the Smith-Waterman algorithm to find matching subsequence. First, test patterns are removed by determining a Euclidean distance with the detector vector T where the detector vector T is computed from the mean hue histogram of frames in the training set. Uniform color frames are detected by computing the entropy of the distribution of color pixels in HSV color space. If an entropy is lower than predefined threshold, it is removed. Clapper boards are detected using a SVM classifier. A training set of 9972 frames labeled as clapper boards, and 15501 frames labeled as non-clapper board are used. For each frame, a feature vector based on the HSV histogram of the central region of the frame is computed and then trained as a SVM classifier using these features. Then, this classifier is used as a clapper board detector. After removing the junk frames, a rushes video is then decomposed into one-second segments. The given segments are clustered using a hierarchical classification that allows tuning the notion of visual similarity by selecting different levels in the hierarchy. Then, a video sequence alignment algorithm based on Smith-Waterman algorithm is used to find repetitive sequences for detecting retakes. However, this method is limited because Smith-Waterman algorithm for sequence alignment and working on frame levels for scene detection are computationally expensive. Moreover it manual adjustment of the threshold for scene detection. A distance measurement approach based on the Longest Common Subsequence (LCSS) algorithm is presented in [20,21], rushes video is decomposed into segment using shot boundary detection based on SVM classifier. The similarity for all shots are computed based on LCSS algorithm. Then, two shots will merge if its length is lager and gab is lower than predefine threshold. Single linkage clustering is used to determine the retakes. However, the use of too many similarity matching based features on LCSS requires a very large computation time.

CHAPTER III

PROPOSED METHOD

3.1 Proposed Framework

The framework of this proposed method shows schematically in Figure 3.1. The input video is first decomposed into shots and then the junk shots are removed. Keyframes are then extracted from each shot and then given frames there are encoded into a string sequence using the location of the object. The LCS algorithm is then enacted to detect the presence of any retakes.

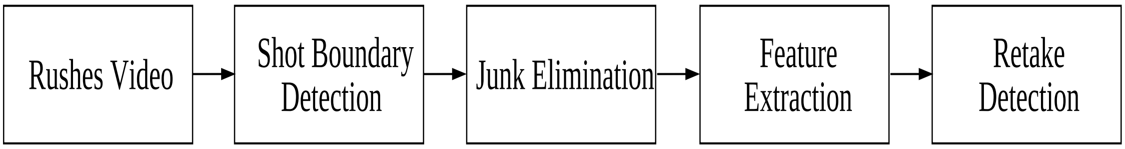


Figure 3.1 Proposed framework.

3.2 Video Representation

The characteristics of a video sequence can be represented by the sequence of objects that appear in the video sequence. Figure 3.2 shows an example of an object moving from left to right. In order to detect the occurrence of objects in video sequence, the object recognition method can be applied. However, the object recognition method cannot determine sufficient information of the object location. In one attempt to overcome this difficulty, an optimal grid size for specifying the location of an object in one or two dimensions by maximizing the information content (entropy) of the system in which the objects reside was introduced [67]. In addition, finding the location of an object by dividing the considerable geographical area into a two dimensional grid has been proposed [68]. In this work, therefore, the object recognition method based on the SIFT feature and the grid method are adopted to determine the object location in the video sequence.

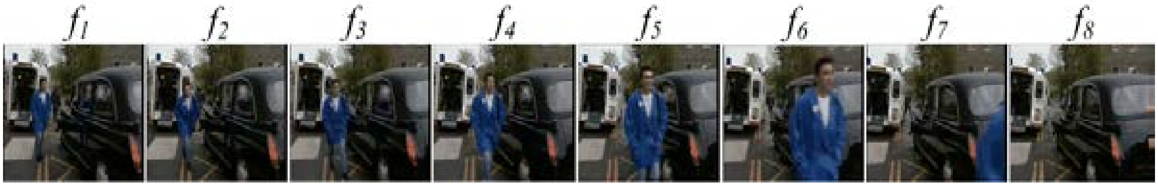


Figure 3.2 An example of object appears in the video sequence.

3.3 Shot Boundary Detection

In order to eliminate redundancy in rushes video, the first needs to organize rushes video into the compact forms or extracts semantically meaningful information. As for the definition of rushes video [25, 26], all rushes are unedited therefore they must consist of hard cuts only. Shot boundary detection is the most basic technique that is wildly used as the first step to organize the video data into segments. In this dissertation, we proposed a new method for hard cut shot boundary detection based on Singular Value Decomposition (SVD) and k-means clustering.

3.3.1 Local Feature Extraction

As for shot boundary detection, a set of features extracted from a frame or a region of frame is an important component. Recently, several studies have already focused on the use of SVD for Image enhancement [41], video sequence matching [42] and video shot boundary detection [43]. By considering the input image as the matrix A, it can be factored into a singular value and the corresponding singular vector factorization. These values are useful information for discriminating image patterns or contents. However, a global feature is very sensitive to motion such as big object and high camera movement. In [59] present a comparison of several shot boundary detection and classification techniques. The results show that a local feature gives high performance than each another. This feature has the advantage of invariants to large object motion and high camera movement. Therefore, the local SVD features are used as the frame representative.

In order to extract a local SVD feature, let frame f_t being an input video frame. Each frame f_t is divided into $B \times B$ blocks. Figure 3.3 shows an example of frame that is divided into 4 × 4 blocks. Let block A be a $M \times N$ matrix of block $b - th$. The SVD of matrix A is then factored into from

$$A = USV^T \tag{3.1}$$

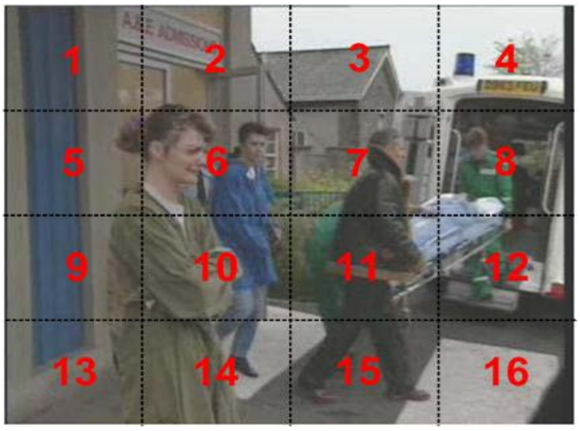


Figure 3.3 An example of frame is divided into 4 ×4 blocks with the layout of local feature.

where U is a $M \times r$ column orthogonal matrix, V is a $N \times r$ column orthogonal matrix, and $S = diag(\sigma_1, \ldots, \sigma_r)$ is a diagonal matrix for $r = min(M, N)$. These diagonal elements are called the singular values (SV). Thus, SV vector can be utilized to describe each blocks $b - th$ of frame f_t effectively. Then, the set of SV vector for every frame f_t are defined as

$$S^T_{b,t} = diag(\sigma_1, \ldots, \sigma_r) \tag{3.2}$$

Where $S^T_{b,t}$ is the SV vector of block $b - th$, $b = 1, 2, \ldots, B^2$. Thus, $S^T_{b,t}[r]$ represents the value of the $r - th$ of the SV vector.

3.3.2 Feature Similarity Measure

In order to detect a shot transition between two adjacent frames based on local feature, an appropriate similarity measure can be used to overcome this. Euclidean distance is probably the most common similarity measure which is used for numerical data. The Euclidean distance between two SV vectors S^T_1, S^T_2 is defined as

$$D_{ecu}(S^T_1, S^T_2) = \sqrt{\sum_{i=1}^{n} (S^T_1[i] - S^T_2[i])^2} \tag{3.3}$$

Thus, the similarity between the $b - th$ block of frames f_t and f_{t+1} can be defined as

$$Dsim(b) = D_{ecu}(S^T_{b,t}, S^T_{b,t+1}) \tag{3.4}$$

An example of the similarity between frame pair is shown in Figure 3.4.

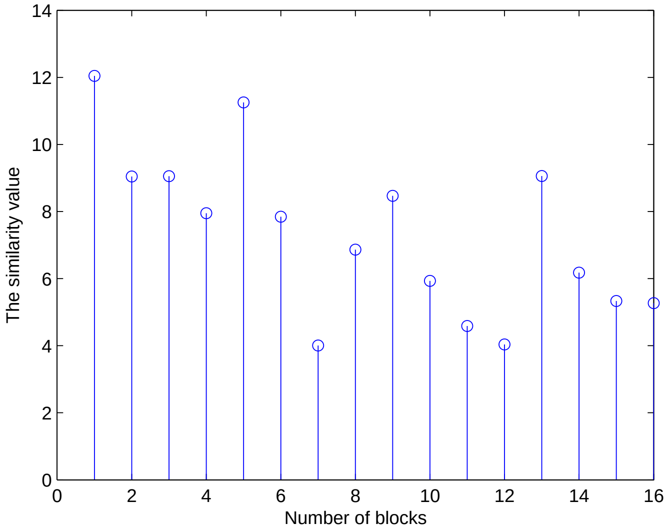


Figure 3.4: Example of the similarity values are computed between frame f_t and f_{t+1} with 4×4 blocks.

3.3.3 Shot Clustering

Usually, most widely methods to detect a shot transition are based on similarity measure between two adjacent frames. A threshold-based can be used to achieve this. The advantages of this method are fast. However, a threshold-based approach cannot achieve equality efficiency for new video data. Hence, more effective methods are needed. The unsupervised clustering can be used to overcome this problem. Then, we employ a simple k-means clustering to detect a shot transition.

As we concentrate on cut boundary detection, then we define a type of boundary into two classes: a normal boundary and cut boundary. A normal boundary is the boundary between two adjacent frames that have the same or nearly visual information. A cut boundary is the boundary between two adjunct frames which have different visual information. Figure 3.5 shows an example two classes of boundaries. Taking this definition into account, we can classify a given feature similarity between each two adjacent frames by using k-means with $k = 2$.

In order to classify a shot boundary, a given $D(sim)[b]$ are sorted into ascending order. Let's b' denote the region index after sorting so that $Dsim(b') \leq Dsim(b' + 1)$. Then, for each two adjacent frames, these values are obtained by sorted their region. According to a problem with large object motion and quick camera movement, this can solve the problem by removing the large values of $Dsim(b')$. Therefore, the values of $Dsim(b')$ for clustering is defined as

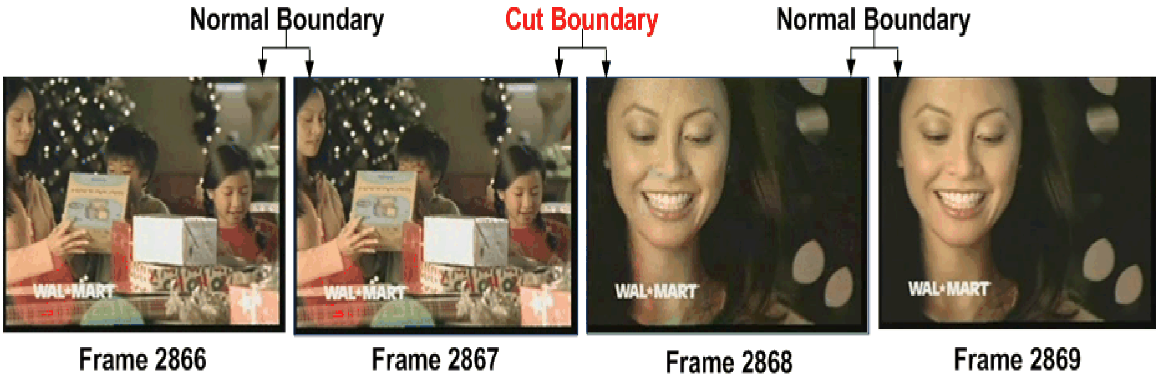


Figure 3.5 Example of two classes boundary (Normal boundary and Cut boundary).

$$x = \sum_{b'=1}^{B^2-\theta} Dsim(b') \tag{3.5}$$

where θ is the number of $Dsim(b')$ that are needed to avoid. The set of input vector is defined as

$$X = (x_1, x_2, \ldots, x_d) \tag{3.6}$$

where d is the dimensionality of X. An example of X is shown in Figure 3.6 and the example of X which plot into the same y coordinate is shown in Figure 3.7. Finally, k-means with $k = 2$ is then apply to X for classify a shot boundary. The example of k-means clustering is shown in Figure 3.8.

3.4 Junk Elimination

By definition, all rushes are unedited and, therefore, they consist of useless shots after shot extraction, such as color bars, single colors, very short shots and clapper boards. In order to reduce the computational time for feature extraction, we need to determine and eliminate these junk shots. After all shot boundaries in the video are detected, the short shots, defined as those that are less than 10 frames in length, are eliminated.

As earlier work [69] has proposed the algorithm to eliminate color bars, single color and clapper boards. This algorithm achieves a higher performance in term of the lack of junk with participating in the TRECVID 2008 summarization task [26]. Therefore, this algorithm is used in this work and more detail of the algorithm are given below.

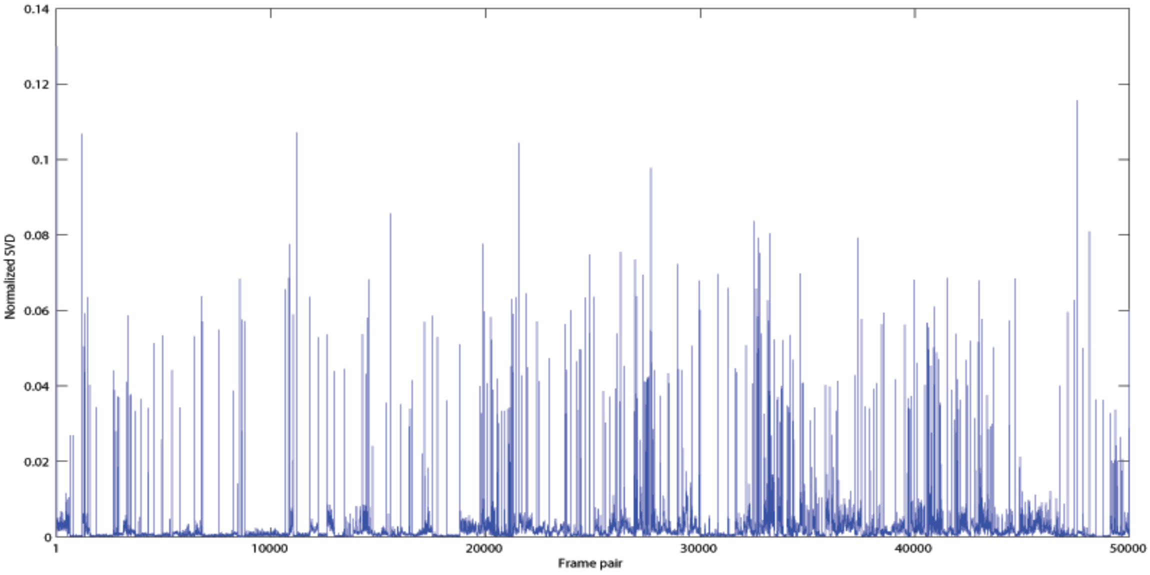


Figure 3.6 Plot of SVD distance between frame pair.

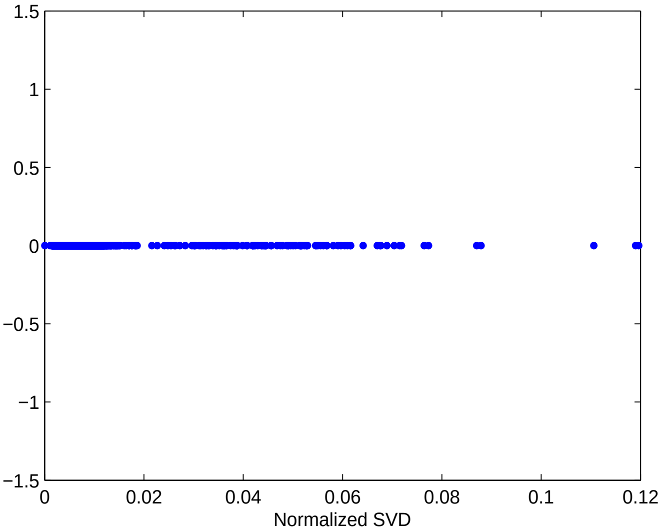


Figure 3.7 Example of normalized SVD.

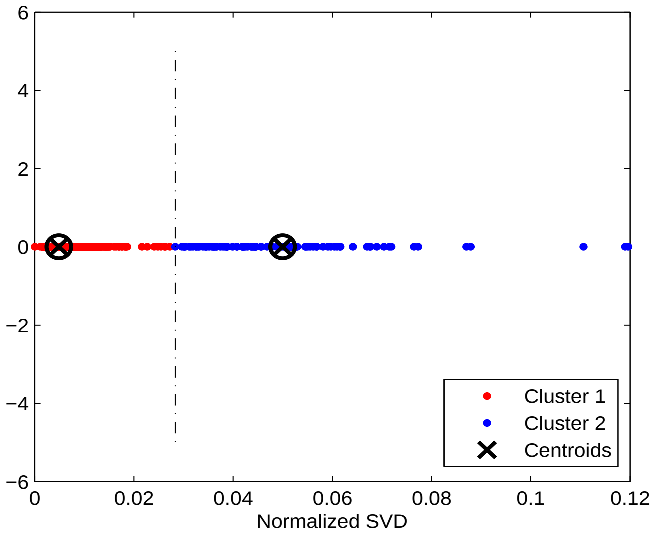


Figure 3.8 Example of clustering result.

3.4.1 Color Bars

As mentioned previously, color bars are useless content which are included for color calibration. Example of color bars are illustrated in Figure 3.9. The characteristics of color bars are vertically averaged, and the color histograms for each block in the same column should be similar. Therefore, a histogram based approach can be adapted for detecting and eliminating color bars.

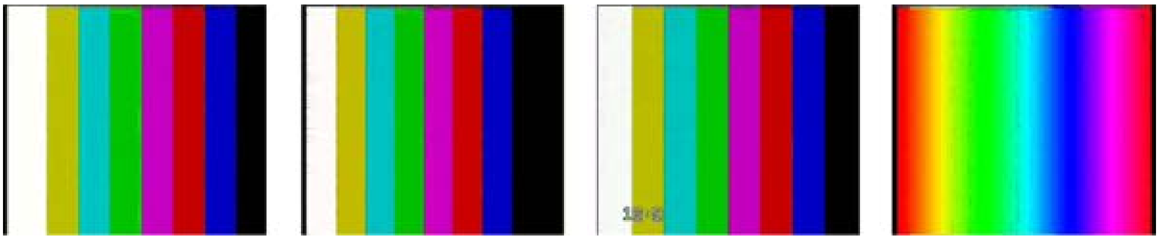


Figure 3.9 Example of color bars are included in rushes video.

Let f_t be a video frame and then a local RGB histogram is extracted by dividing frame f_t into 4×4 blocks. A 64-bin histogram is used for each channel. Let H_k^R , H_k^G and H_k^B are denoted the local color histogram for k-th block of frame f_t, where $k = 1..16$. Hence, $H_k^R(i)$ is represented the value of the i-th bin of the R histogram, where $i = 1..64$.

In order to detect color bars, the χ^2 distance is used to compute the histogram differences between any two neighboring blocks in each column. In the case of the R channel, the χ^2 distance between two histogram $H1^R$ and $H2^R$ are defined as

$$D_{\chi^2}(H1^R, H2^R) = \sum_{i=1}^{n} \begin{cases} \frac{(H1^R(i)-H2^R(i))^2}{max(H1^R(i),H2^R(i))} & \text{if } max(H1^R(i), H2^R(i)) > 0\,, \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

where n is the number of the bin histogram. Hence, the χ^2 distance between any two neighboring block in each column are defined as

$$D^R_{nb}(k) = \begin{cases} D_{\chi^2}(H1^R_k, H2^R_{k+4}) & \text{if } k \leq 13\,, \\ D_{\chi^2}(H1^R_k, H2^R_{k-12}) & \text{otherwise.} \end{cases} \tag{3.8}$$

where $k = 1 \dots 16$. Then, the total χ^2 distance between any two neighboring block is defined as

$$D_{nb}(k) = \frac{1}{3}(D^R_{nb}(k) + D^G_{nb}(k) + D^B_{nb}(k)) \tag{3.9}$$

Then, a given $D_{nb}(k)$ are sorted into ascending order. Let k' denote the index after sorting so that $D_{nb}(k') \leq D_{nb}(k'+1)$. Therefore, if the value of 10^{th} of $D_{nb}(k')$ is smaller than the threshold θ_{cb}, then frame f_t is defined as a color bar frame.

3.4.2 Single Color frames

In rushes videos, a single color is includes not only for camera calibration but also supports to the measurement illumination level. An example of single color is illustrated in Figure 3.10. From the properties of single color image, a dominant color in its global histogram is large. Therefore, a single color can be detected by using a global RGB histogram. If these value are large than a predefine threshold, then the given frame is defined as a single color.

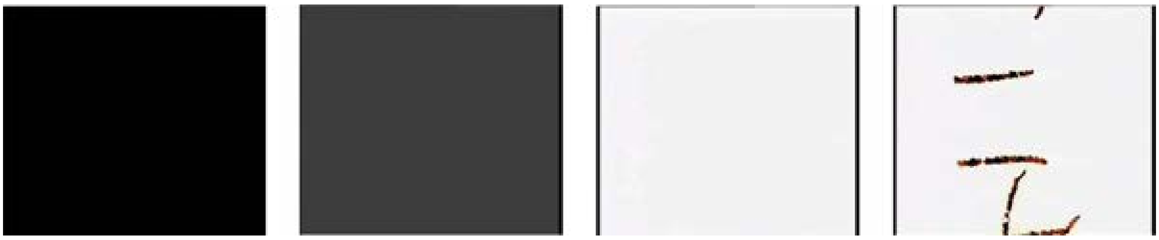


Figure 3.10 Example of single color are included in rushes video.

Let f_t be a video frame and then a local RGB histogram is extracted by dividing frame f_t into 4×4 blocks. A 64-bin histogram is used for each channel. Let H^R_k , H^G_k and H^B_k are denoted the local color histogram for the k-th block of frame f_t, where $k = 1..16$. Hence, $H_k^R(i)$ is represented by the value of the i-th bin of the R histogram, where $i = 1..64$.

Next, determine a sum of average of the i-th bin of the RGB histogram. The H_μ of the i-th bin is defined as

$$H_\mu(i) = \frac{1}{C \times n} \sum_{k=1}^{n} \left(H_k^R(i) + H_k^G(i) + H_k^B(i) \right) \tag{3.10}$$

where C is the number of the color channel, n is the number of blocks, and i is the i-th bin of histogram. Therefore, if max$(H_u(i))$ is larger than the threshold θ_{single} then frame f_t is defined as a single color.

3.4.3 Clapper Boards

In rushes videos, there are many types of clapper board appearances but the same type of clapper boards are often used in the same movie. The clapper boards have many types such as scale, rotation and illumination changes. Example of clapper boards are illustrated in Figure 3.11 and Figure 3.12. The NDK algorithm, proposed in [56] is invariant to image scaling, translation, rotation, illumination changes and affine or 3D projection. Therefore, this algorithm is sufficient for detection.

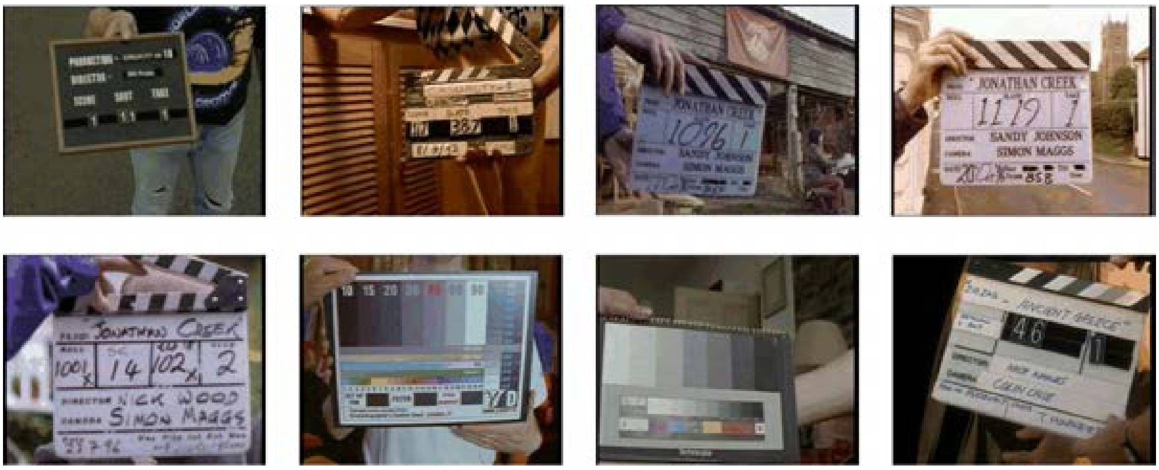


Figure 3.11 Example of clapper boards are included in rushes video.

To detect a clapper boards, a set of 80 example frames of the clapper board are extracted from the TRECVID development set. The backgrounds where clapper boars are not present were manually removed. The key points are extracted for each frame. Then, the given features are used as a set of queries. The step for using NDK algorithm to detect a clapper boards are as following:

- Let f_t be a frame which is extracted from input video and then, extracted the key points.

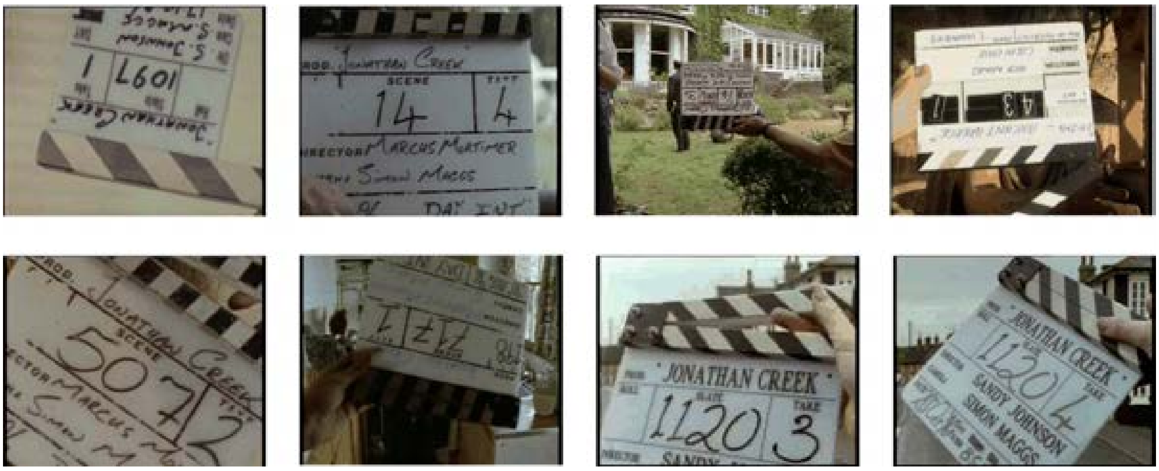


Figure 3.12: Examples of clapper boards appear in many situation such as scale, rotation and illumination changes.

- The given key points are matched with the set of queries by using NDK algorithm.
- If the result of the NDK algorithm returns out of match from the given key points with the query then the frame f_t is defined as a clapper board frame.

3.5 Feature Extraction

As mentioned previously, a video sequence can be represented by the sequence of objects which appear in the video. The object recognition method can be applied to detect the occurrence of objects in video sequence. However, the object recognition method cannot determine sufficient information of the object location. Therefore, in this work, the object recognition method based on the SIFT feature and the grid method are adopted to determine the objection location.

Let a set of shots $V = (S_1, S_2, ..., S_n)$, where n is number of shots. Each shot is represented by a set of key frames. In order to reduce the computation time, the key frames are extracted from the original video at every 10^{th} frame. Each frame is divided into $b \times b$ blocks. From our experiment, $b = 5$ provided the best result. Let s_k denote the SIFT feature for the k-th block of frame f_t. In the standard SIFT features, each SIFT feature is represented as $s_k[i] = (d_i, \sigma_i, x_i, y_i)$, where $i = 1...N$, N is the number of SIFT features detected on block k-th, d_i is the 128 dimensional SIFT descriptor, σ_i is the scale of SIFT features. x_i and y_i are the SIFT feature location.

Then, s_k of each block, given as defined below, are extracted based on [48–51] (see Figure 3.13). The set of s_k are given as

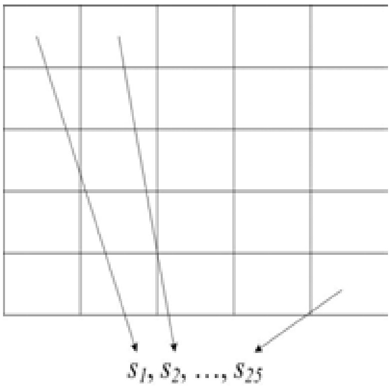


Figure 3.13 A frame is divided into 5×5 blocks.

$$f_t = (s_1, s_2, ..., s_{25}) \tag{3.11}$$

where f_t is a frame i^{th}, s_1 is the SIFT features that are extracted from block 1.

In order to select the k-th blocks as object location representative, SIFT matching [55] can be used to find the number of SIFT features that can be considered to match. To determine the SIFT features are matched or not, the squared Euclidean distance and nearest neighbor algorithm are performed. First, the distance between SIFT features are computed by using squared Euclidean distance in the descriptor space. The squared Euclidean distance $D^2(s[1], s[2])$ between two SIFT features $s[1]$ and $s[2]$ is defined as

$$D^2(s[1], s[2]) = \sum_{i=1..128} (d_1[i] - d_2[i])^2 \tag{3.12}$$

where d_1 is a SIFT descriptor of SIFT feature $s[1]$, and d_2 is a SIFT descriptor of SIFT feature $s[2]$.

Second, the SIFT features $s[1]$ and $s[2]$ are considered matched if the distance ratio between the nearest neighbor distance and the second nearest neighbor distance is below τ^2 ,

$$\frac{D^2(s[1], s[2])}{D^2(s[1], \acute{s}[3])} < \tau^2 \tag{3.13}$$

where $D^2(..., ...)$ is the squared Euclidean distance between two SIFT descriptors, $\acute{s}[3]$ is the second nearest SIFT feature of $s[2]$ and τ is a threshold for determining whether $s[1]$ and $s[2]$ are matched or not. The number of SIFT matching result in the same block between frame f_t and f_{t+1} is implemented as follows:

Algorithm for determining the number of SIFT matching result

```
Input: $M$ = number of SIFT features in block $k-th$ of frame $f_t$
       $N$ = number of SIFT features in block $k-th$ of frame $f_{t+1}$
Output: $\mu_k$ = number of matched result
Initialize: $\mu_k = 0$
            $\tau = 0.8$
for $i = 1$ to $M$ do
    distA = $\infty$
    distB = $\infty$
    for $j = 1$ to $N$ do
        dist = $D^2(s_{k,t}[i], s_{k,t+1}[j])$
        if dist $<$ distA then
            distB = distA
            distA = dist
        else if dist $<$ distB then
            distB = dist
        end if
    end for
    if (distA/distB) $< \tau^2$ then
        $\mu_k = \mu_k + 1$
    end if
end for
```

After performing the algorithm, the set of μ_k features between frame f_t and f_{t+1} are given as

$$M = (\mu_1, \mu_2, \ldots, \mu_k) \tag{3.14}$$

where $M(k)$ is the set of SIFT matching results between frame f_t and f_{t+1}, and μ_1 is SIFT matching result between frames f_t and f_{t+1} at block 1. The example is shown in Figure 3.14 and 3.15.

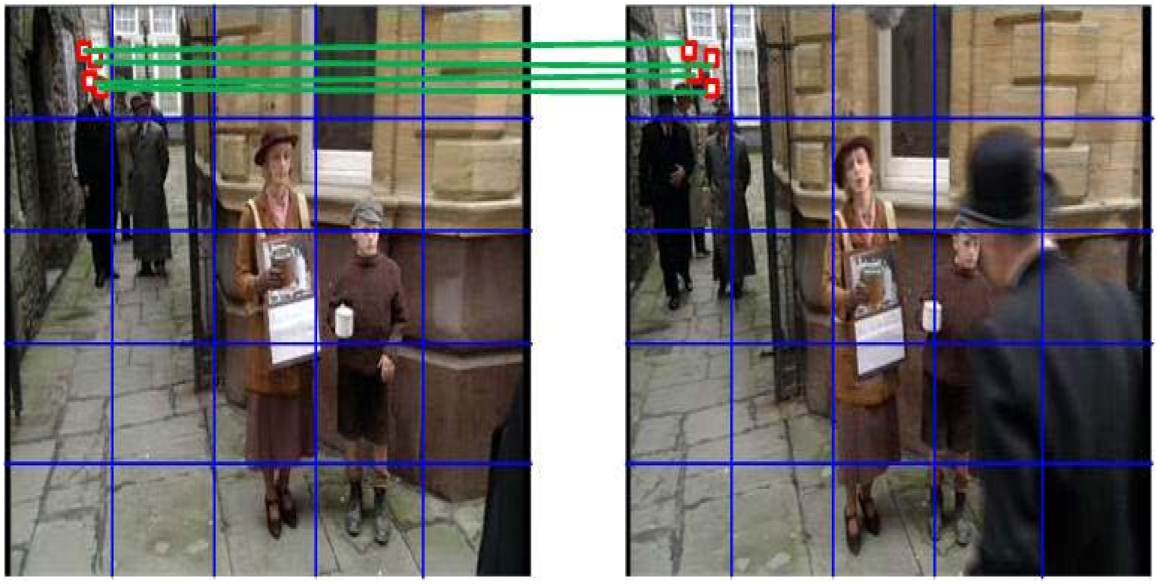
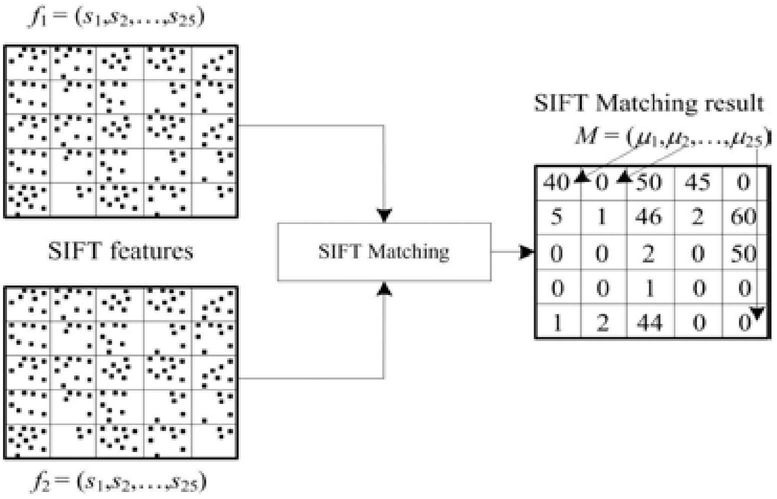


Figure 3.14 An example of SIFT matching result is matched between frames f_t and f_{t+1} at block1.


$f_1 = (s_1, s_2, \ldots, s_{25})$
SIFT features
$f_2 = (s_1, s_2, \ldots, s_{25})$
SIFT Matching
SIFT Matching result
$M = (\mu_1, \mu_2, \ldots, \mu_{25})$

40	0	50	45	0
5	1	46	2	60
0	0	2	0	50
0	0	1	0	0
1	2	44	0	0

Figure 3.15 An example of SIFT matching result.

To select k-th blocks as the object location representative, a threshold based is performed. Let Th_{select} denote the threshold for select the k-th blocks. The Th_{select} is defined as

$$Th_{select} = \frac{\alpha}{n} \sum_{k=1}^{n} M(k) \tag{3.15}$$

where n is the number of k-th blocks and α is constant. If the value of μ_k is over than the threshold Th_{select}, then block at k-th is selected. The example is shown in Figure 3.16.

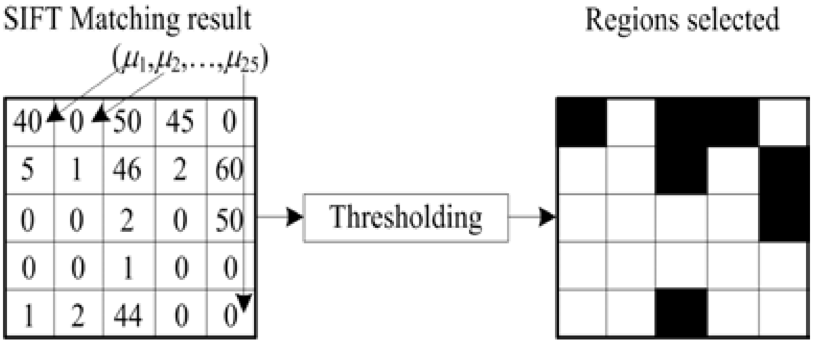


Figure 3.16 An example of blocks selection.

In order to encode the set of selected features into a string sequence, the string representation approach is performed. Let a set of English alphabet corresponds to the grid blocks (see Figure 3.17). The sequence of string is determined by matching the index of the selected features. For example, if the selected features are $1, 2, 3$ and 5, it means that the similarity between frame f_t and frame f_{t+1} are matched at blocks 1, 2, 3 and 5 (For an example see Figure 3.18).

A 1	B 2	C 3	D 4	E 5
F 6	G 7	H 8	I 9	J 10
K 11	L 12	M 13	N 14	O 15
P 16	Q 17	R 18	S 19	T 20
U 21	V 22	W 23	X 24	Y 25

Figure 3.17 A frame is divided into 5×5 blocks and the set of English alphabet are assigned.

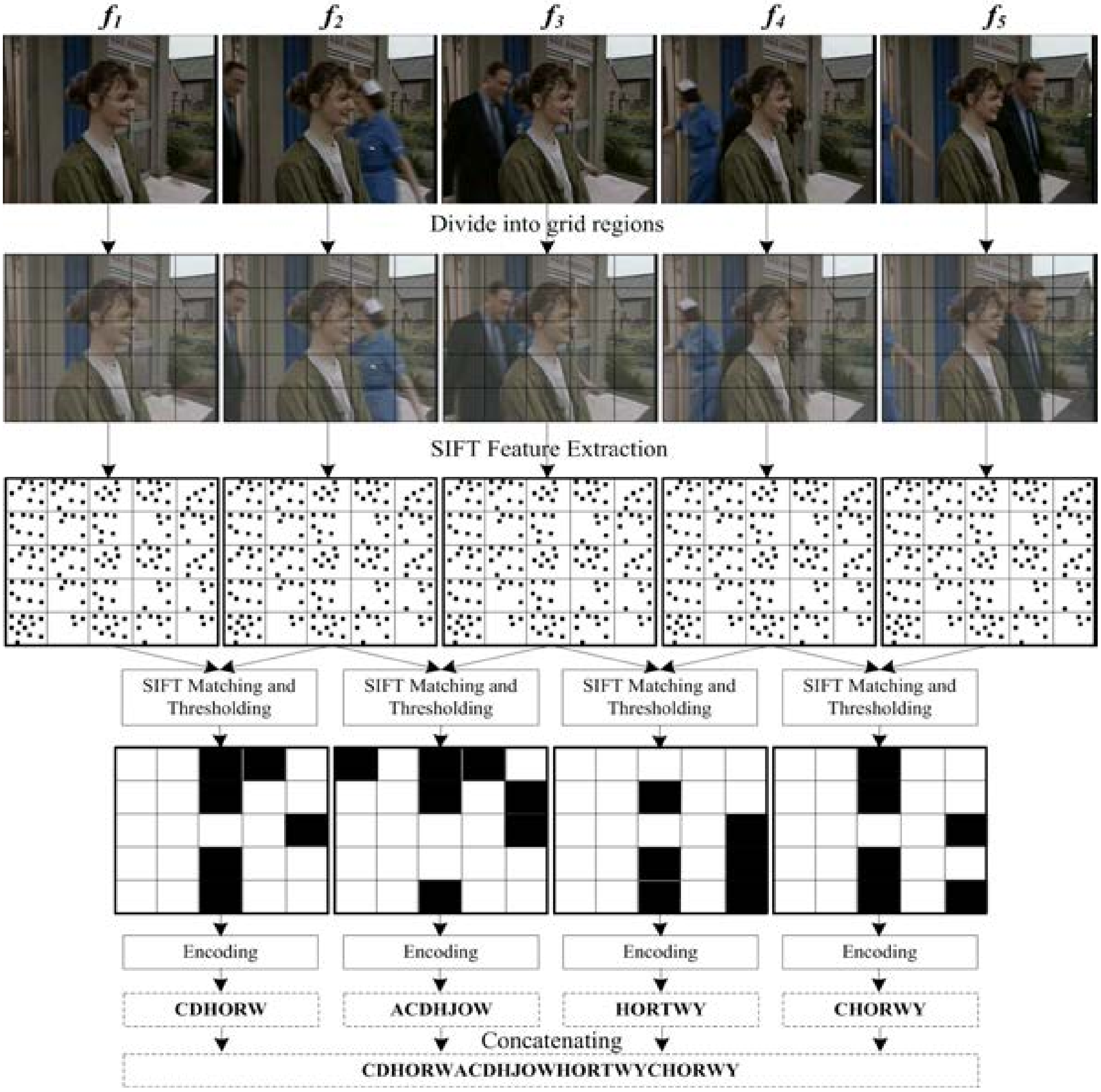

f_1
f_2
f_3
f_4
f_5
Divide into grid regions
SIFT Feature Extraction
SIFT Matching and Thresholding
SIFT Matching and Thresholding
SIFT Matching and Thresholding
SIFT Matching and Thresholding
Encoding
Encoding
Encoding
Encoding
CDHORW
ACDHJOW
HORTWY
CHORWY
Concatenating
CDHORWACDHJOWHORTWYCHORWY

Figure 3.18 An example of five frames that are encoded into a string sequence.

3.6 Retake Detection

The characteristics of takes that appear in a scene are schematically shown in Figure 3.19. The repeated take of the same scene appears as a sequence of order. In order to detect the repeated take in the same scene, the retake detection algorithm is implemented as follows:

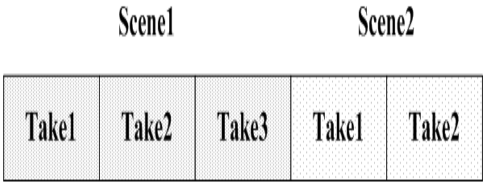


Figure 3.19 An example of takes that appear in the two scenes.

Algorithm 1 The algorithm for retake detection.

```
Input: $V = \{shot_1, shot_2, \ldots, shot_n\}$ (set of shots to be detection)
        $l$ (number of shots)
Output: $S = \{s_1, s_2, \ldots, s_n\}$ (set of scene)
        $s = \{t_1, t_2, \ldots, t_n\}$ (set of take)
Initialize: $l = length(V)$
if $l \geq 2$ then
        $k = 1, j = 1$
        $t_j = shot_k$
        $s_k = t_j$ (create a new scene and add the first shot)
        for $i = 2$ to $l$ do
                $\mu = LCS(s_k, shot_i)$ (determine subsequence between two shots)
                if $\mu \geq Threshold$ then
                        $j = j + 1$
                $t_j = shot_i$
                $s_k = s_k \cup t_j$ (add the repeated shot to the scene)
            **else**
                $k = k + 1$
                $j = 1$
                $t_j = shot_i$
                $s_k = s_k \cup t_j$ (create a new scene and add a new shot)
            **end if**
        **end for**
**end if**
```

The threshold from algorithm one is defined as

$$Threshold = \theta min(L_i, L_j) \tag{3.16}$$

where L_i is length of sequence which extracts from $Shot_i$, L_j is length of sequence which extracts from $Shot_j$, and θ is constant. Figure 3.20 shows an example of two shots whose common subsequence is determined by using the LCS algorithm.

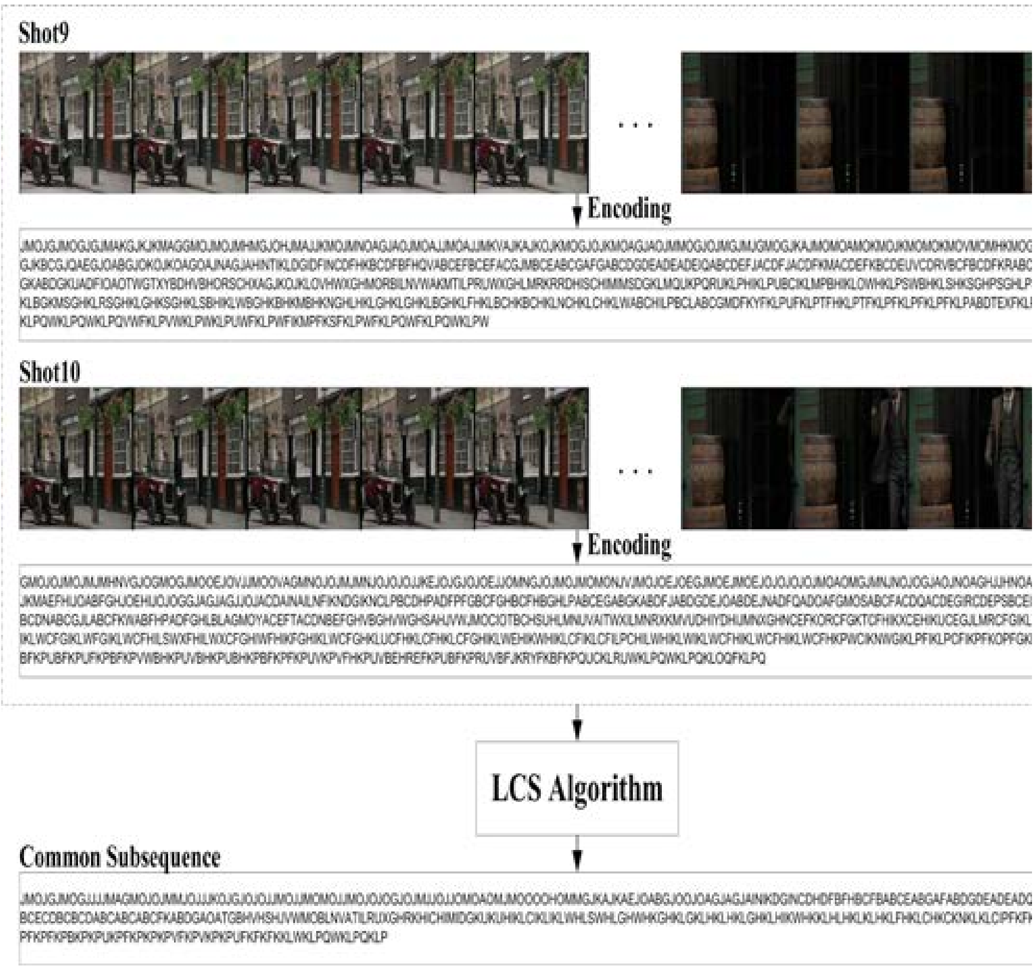


Figure 3.20 An example of common subsequence is determined from shot9 and shot10.

CHAPTER IV

EXPERIMENTAL RESULTS

4.1 Data sets

4.1.1 Shot Boundary Detection

All experiments of shot boundary detection are tested and evaluated on TRECVID 2004-2007 data set [58]. This data set have varied widely from English broadcast TV news (ABC& CNN), Arabic TV news, Chinese TV news, Sound&Vision educational, news magazine and historical. Some of the video characteristics are shown in Table 4.1.

Table 4.1 TRECVID shot Boundary detection data set.

Year	Hrs.	Files	Frames	Trans	%Cut	%Gradual	Data description
2004	6.0	12	618,409	4,806	57.7	43.3	English broadcast TV news (ABC & CNN)
2007	6.0	17	637,805	2,317	90.8	9.2	Sound & Vision educational, news magazine, historical

4.1.2 Junk Elimination, Feature Extraction and Retake Detection

TRECVID 2007-2008 rushes summarization data set [1, 2] are used for experiments of retake detection. 14 of these data sets are selected for testing and were evaluated. Some of the video characteristics are show in Table 4.2. These data sets consisted of unedited video footage, shot mainly for five series of BBC drama programs. The drama series included a historical drama set in the early 1900's, a series on ancient Greece, a contemporary detective program, a program on emergency services, and a police drama. Rushes are contained scenes of people in various everyday situations, both indoor and outdoor. Some actors appeared repeatedly in the same setting and in other settings. There redundancy consisted from the scenes that were shot and then re-shot with the camera runs. The crew and clapper boards appeared in the scenes and at take boundaries.

Table 4.2 Experimental rushes video for tested and evaluated.

Name of videos	Frames	Frame rate(fps)	Frame size(pixels)
MRS150072	34,382	29.97	352×288
MRS025913	38,567	29.97	352×288
MRS044500	32,058	29.97	352×288
MRS145918	14,141	29.97	352×288
MRS035126	46,427	29.97	352×288
MRS044499	18,587	29.97	352×288
MRS044725	42,873	29.97	352×288
MRS045104	47,387	29.97	352×288
MRS145332	37,653	29.97	352×288
MRS145343	27,767	29.97	352×288
MRS148797	42,077	29.97	352×288
MRS151099	42,827	29.97	352×288
MRS151585	24,858	29.97	352×288
MRS146570	45,313	29.97	352×288

4.2 Performance Evaluation

A good shot boundary detection or retake detection should minimize the number of false detections while maximizing the number of correctly identified shot boundary or retake. The three measures, recall, precision and F1 are usually used for detection and retrieval problem. Recall indicates the proportion of relevant material that is retrieved, while precision is a measure of how relevant the retrieved or selected information is correct. F1-measure is the weighted harmonic mean of precision and recall. Then, recall, precision and F1 are computed based on the following equation:

$$Recall = \frac{Correct}{Correct + Missed},$$

$$Precision = \frac{Correct}{Correct + False}, \tag{4.1}$$

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}.$$

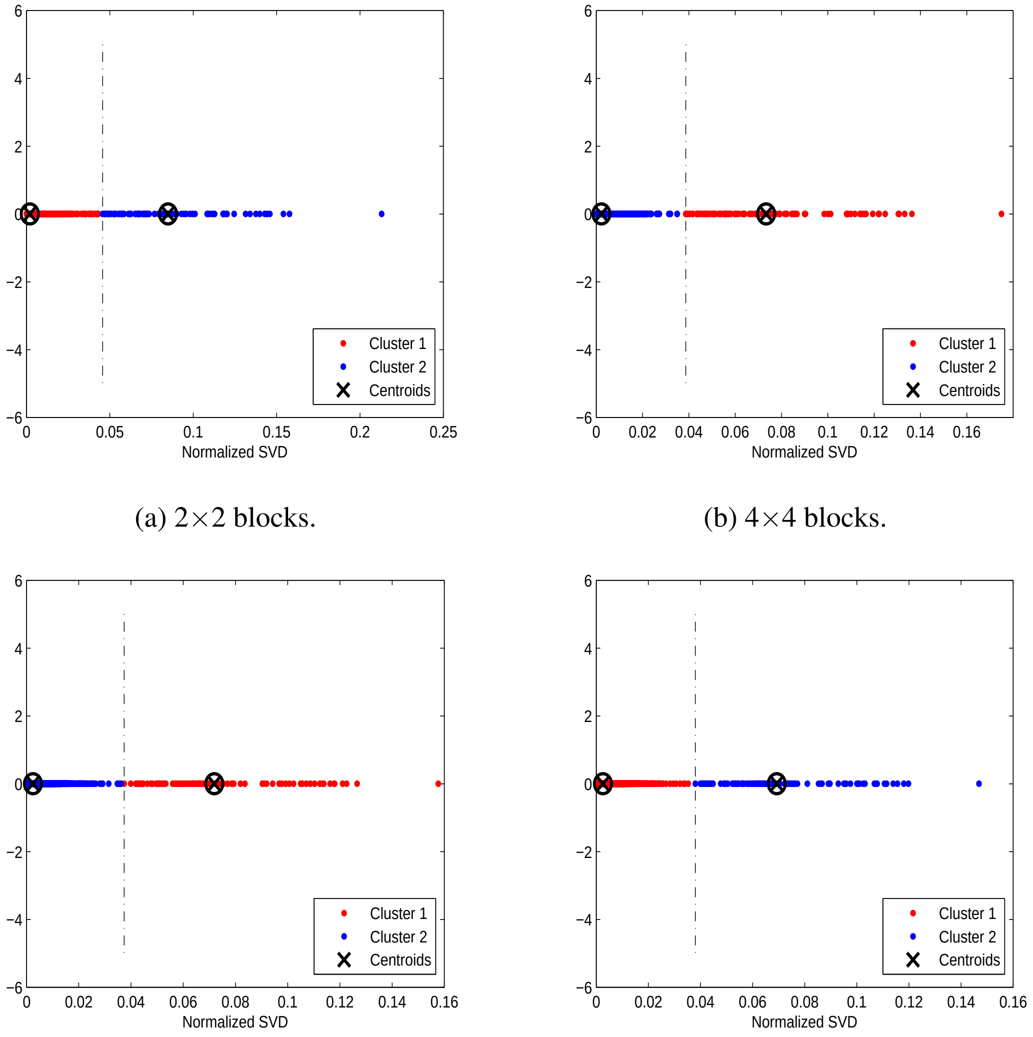
where $Correct$ is the number of correctly retrieved shot boundary or retake, $Missed$ is the number of missed retrieved shot boundary or retake, $False$ is the number of false retrieved shot boundary or retake. Recall and precision jointly rate performance of a classification/retrieval technique and a successful method produces recall and precision values which are close to unity.

4.3 Shot Boundary Detection

4.3.1 Comparison for number of block selection

This experiment is used to compare the performance on a number of block that provides the best result for shot boundary detection. The data sets used in this experiment are obtained from TRECVID 2007 data set that consisted of eight videos. The videos are in MPEG-1 format with frame rate of 29.97 fps and the frame size of 352×288 pixels. The ground truth provided by TRECVID was used for evaluating the results. In order to compare the performance on a number of block, the input videos are divided into $B \times B$ blocks where $B = \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. The local SVD feature is then extracted from each block. The similarity between each two adjacent frames is determined by using Euclidean distance. The set of input vectors are created by using equation 3.5 and 3.6 with $\theta = 0$. Then, k-means with $k = 2$ is applied to the set of input vector to classify a shot boundary. The clustering results of video BG_2408 is shown in Figure 4.1. The performance comparison is shown in Table 4.3.

From the results as shown in Table 4.3, the block size of 8×8 and 10×10 provide a higher recall value than a block size of 2×2, 3×3, 4×4, 5×5, 6×6, 7×7 and 9×9 , but a block size of 2×2, 4×4 and 5×5 provide a higher precision value than the block size of 8×8 and 10×10. It implies that the small number of blocks has missed detecting than the large number of blocks. Because it cannot be differentiated between shot boundaries that have similar visual contents. However, the large number of blocks has false detecting than the small number of blocks. Due to the fact that it cannot differentiate between hard cuts and the large object motion or quick camera movement. Then in this work, a block size of 8×8 was chosen empirically as it gives a higher recall and precision value than a block size of 10×10.



(a) 2×2 blocks.

(b) 4×4 blocks.

(c) 6×6 blocks.

(d) 8×8 blocks.

Figure 4.1: Clustering results of video BG_2408. The left side of vertical line is a normal boundary, and the right side of vertical line is a cut boundary.

4.3.2 Comparison for number of large value removal

From the previous experiment, the large number of blocks cannot differentiate between hard cuts and the large object motion or quick camera movement. This can be overcome the problem by remove the large similarity value of local SVD feature between two adjacent frames. The purpose of

Table 4.3 Performance comparison of number of blocks selection.

Video name	**Recall**								
	2×2	**3×3**	**4×4**	**5×5**	**6×6**	**7×7**	**8×8**	**9×9**	**10×10**
BG_2408	0.80	0.95	0.99	0.99	0.99	0.99	1.00	0.99	1.00
BG_37359	0.85	0.90	0.95	0.96	0.96	0.96	0.98	0.97	0.98
BG_35050	0.87	0.93	0.96	0.96	0.97	0.96	0.97	0.96	0.97
BG_36028	0.92	0.92	0.97	0.99	0.99	0.99	0.99	0.99	0.99
BG_37417	0.93	0.97	0.99	0.99	1.00	0.99	1.00	0.99	0.99
BG_35187	0.74	0.79	0.87	0.89	0.95	0.93	0.95	0.95	0.95
BG_36537	0.85	0.90	0.91	0.92	0.92	0.92	0.92	0.93	0.93
BG_37879	0.91	0.94	0.94	0.95	0.98	0.97	0.99	0.99	0.99
Average	0.86	0.91	0.95	0.96	0.97	0.96	**0.98**	0.97	**0.98**

Video name	**Precision**								
	2×2	**3×3**	**4×4**	**5×5**	**6×6**	**7×7**	**8×8**	**9×9**	**10×10**
BG_2408	0.94	0.94	0.93	0.93	0.96	0.95	0.96	0.96	0.96
BG_37359	0.99	0.97	0.95	0.93	0.90	0.90	0.90	0.88	0.86
BG_35050	0.98	0.97	0.97	0.98	0.97	0.98	0.98	0.98	0.98
BG_36028	0.81	0.81	0.88	0.89	0.87	0.86	0.85	0.86	0.85
BG_37417	0.97	0.95	0.97	0.94	0.92	0.91	0.90	0.90	0.90
BG_35187	0.97	0.99	0.99	0.99	0.98	0.98	0.98	0.97	0.97
BG_36537	0.93	0.90	0.90	0.90	0.89	0.90	0.89	0.88	0.87
BG_37879	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Average	**0.95**	0.94	**0.95**	**0.95**	0.94	0.94	0.93	0.93	0.92

Video name	**F1**								
	2×2	**3×3**	**4×4**	**5×5**	**6×6**	**7×7**	**8×8**	**9×9**	**10×10**
BG_2408	0.86	0.94	0.96	0.96	0.97	0.97	0.98	0.98	0.98
BG_37359	0.91	0.93	0.95	0.94	0.93	0.93	0.94	0.92	0.91
BG_35050	0.92	0.95	0.96	0.97	0.97	0.97	0.97	0.97	0.97
BG_36028	0.86	0.86	0.92	0.94	0.93	0.92	0.91	0.92	0.91
BG_37417	0.95	0.96	0.98	0.96	0.96	0.95	0.95	0.94	0.94
BG_35187	0.84	0.88	0.93	0.94	0.96	0.95	0.96	0.96	0.96
BG_36537	0.89	0.90	0.90	0.91	0.90	0.91	0.90	0.91	0.90
BG_36537	0.95	0.97	0.97	0.97	0.99	0.98	0.99	0.99	0.99
Average	0.90	0.92	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**	**0.95**

this experiment is used to compare the performance on remove a number of large similarity value of local SVD feature that provides the best result for shot boundary detection. The data sets used in this experiment are obtained from TRECVID 2007 data set that consisted of eight videos. The videos are in MPEG-1 format with frame rate of 29.97 fps and the frame size of 352×288 pixels. The ground truth provided by TRECVID was used for evaluating the results. To compare the performance on remove a number of large value, the input videos are divided into 8×8 blocks that provides the best performance from the previous experiment. The local SVD feature is then extracted from each block. The similarity between each two adjacent frames is determined by using Euclidean distance. The set of input vectors are created by using equation 3.5 and 3.6 with the parameter θ is varied between 0 to 63. Then, k-means with $k = 2$ is applied to the set of input vector for classify a shot boundary. The performance comparison is shown in Table 4.4.

From the results, the low value of parameter θ gives the high recall and precision on average. Because the large object motion or quick camera movement could result in very different intensity distributions for the blocks affected by object motion and camera movement. When these blocks have removed, thus it reduces the impact from the large object motion or quick camera movement. The high value of parameter θ gives the high precision, but it gives low recall value on average. Due to it rejected too many of shot boundary.

Then in this work, the value of parameter θ between 5 to 9 was chosen empirically as it gives the better compromise between recall and precision.

4.3.3 Comparison for performance

In this experiment is used to compare a performance of the proposed method and other method. The method based on Support Vector Machine (SVM) by Le [63] provides the best performance in cut detection with TRECVID 2003 data set. Therefore, to evaluate the shot boundary detection performance, the proposed method is compared with Le [63]'s method by setting the parameter as recommended in [63]. The data sets used in this experiment are obtained from TRECVID 2004 and 2007 data set that consisted of 12 videos. Le [63]'s method is trained by using four videos from the TRECVID 2004. The eight videos from TRECVID 2007 are used to compare a performance. The videos are in MPEG-1 format with frame rate of 29.97 fps and the frame size of 352x288 pixels. The ground truth provided by TRECVID was used for evaluating the results. For the proposed method, the input videos are divided into 8×8 blocks, and then extracted the local SVD feature form each block. The similarity between each two adjacent frames is determined by using Euclidean distance. The set of input vectors are created by using equation 3.6 and 3.7 with the parameter $\theta = 5$. Then,

Table 4.4 Performance comparison for the parameter θ is varied between 0 to 63.

The value of parameter θ	**The average of recall**	**The average of precision**	**The average of F1**	**The value of parameter θ**	**The average of recall**	**The average of precision**	**The average of F1**
0	0.98	0.93	0.95	32	0.92	0.98	0.95
1	0.97	0.95	0.96	33	0.92	0.98	0.95
2	0.97	0.95	0.96	34	0.92	0.98	0.95
3	0.97	0.95	0.96	35	0.92	0.98	0.95
4	0.97	0.95	0.96	36	0.92	0.98	0.95
5	**0.96**	**0.96**	**0.96**	37	0.92	0.98	0.95
6	**0.96**	**0.96**	**0.96**	38	0.91	0.98	0.94
7	**0.96**	**0.96**	**0.96**	39	0.90	0.98	0.94
8	**0.96**	**0.96**	**0.96**	40	0.90	0.98	0.94
9	**0.96**	**0.96**	**0.96**	41	0.89	0.98	0.93
10	0.95	0.97	0.96	42	0.89	0.98	0.93
11	0.95	0.96	0.95	43	0.89	0.98	0.93
12	0.95	0.97	0.96	44	0.88	0.98	0.93
13	0.95	0.97	0.96	45	0.88	0.98	0.93
14	0.95	0.97	0.96	46	0.88	0.98	0.93
15	0.95	0.97	0.96	47	0.87	0.98	0.92
16	0.94	0.97	0.95	48	0.87	0.98	0.92
17	0.94	0.97	0.95	49	0.87	0.98	0.92
18	0.94	0.97	0.95	50	0.85	0.98	0.91
19	0.94	0.98	0.96	51	0.84	0.99	0.91
20	0.94	0.98	0.96	52	0.80	0.99	0.88
21	0.93	0.97	0.95	53	0.80	0.99	0.88
22	0.93	0.97	0.95	54	0.79	0.99	0.88
23	0.93	0.97	0.95	55	0.79	0.99	0.88
24	0.93	0.98	0.95	56	0.77	0.99	0.87
25	0.93	0.98	0.95	57	0.76	0.99	0.86
26	0.92	0.98	0.95	58	0.75	0.99	0.85
27	0.92	0.98	0.95	59	0.74	0.99	0.85
28	0.92	0.98	0.95	60	0.73	0.98	0.84
29	0.92	0.98	0.95	61	0.70	0.98	0.82
30	0.92	0.98	0.95	62	0.66	0.98	0.79
31	0.92	0.98	0.95	63	0.58	0.97	0.73

k-means with $k = 2$ is applied to the set of input vector for classify a shot boundary. The performance comparison is show in Table 4.5.

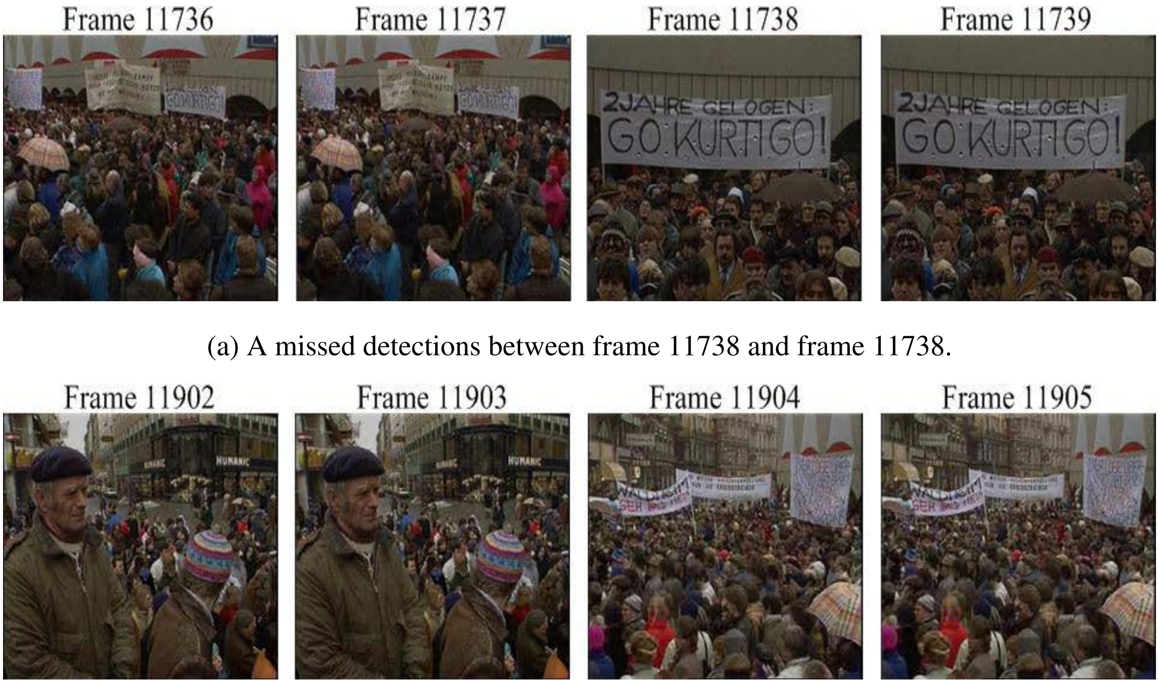
From the result shown that the proposed method has recall rate similar to Le [63], but it gives much better precision rate on average. For video BG_36028, BG_37417 and BG_37879, the proposed method has better recall and precision rate. For video BG_37359 and BG_36537, Le [63] has a better recall and precision rate. Video BG_35187 gives a low recall rate with the proposed method, but it gives a better precision rate. Due to this video has similar backgrounds and visual content in some adjacent shots. An example of missed detections by the proposed method is shown in Figure 4.2. On the another hand, Le [63] has a low recall and precision rate. An example of missed and false detection by Le [63] is shown in Figure 4.3. From the Figure 4.3(b) shown that, Le [63] is very sensitive to small changes. In addition, it cannot detect a shot boundary that is not included in the training set as shown in Figure 4.3(a). For video BG_37879, Le has a better precision rate similar to the proposed method, but it gives a low recall rate. An example of missed and false detections by Le [63] is shown in Figure 4.5. From the Figure 4.5(a) shown that, Le [63] cannot detect an adjacent shot that has a blur background. Moreover, it is very sensitive to small changes in an adjacent shot as shown in Figure 4.5(b). An example of missed and false detection by the proposed method is shown in Figure 4.4. From the Figure 4.4(a) shown that, the proposed method cannot detect an adjacent shot due to it has a similar background. From the Figure 4.4(b) shown that, the proposed method has been false detect an adjacent shot because it cases an effected by intensity change. From this experimental result shown that, the proposed method has a better performance to detect shot boundary than Le [63].

4.4 Junk Elimination

In this experiment, the method proposed in [69] is used to extract keyframes from input video which use to test an accuracy of color bars removal, single color removal and clapper boards removal. First, The shot boundary detection algorithm in [8] is used to define a boundary and partition a video into shots. A local color histogram is extracted by dividing a video frame into 4×4 blocks. The χ^2 distance is used to compute the distance between each blocks of frames f_t and f_{t+1}. Next, these values were sorted into an ascending order. The sum of the middle eight of these 16 values are used to define a cut between frames f_t and f_{t+1} if these values exceed a threshold Th_{shot}. However, this algorithm cannot distinguish between hard cut and the large objects motion. To overcome this problem, motion-based features are computed for each video frame using the Lucas-Kanade point-based tracking functions included in the OpenCV. The magnitude is computed from the motion vector

Table 4.5 Performance comparison of proposed method and existing method.

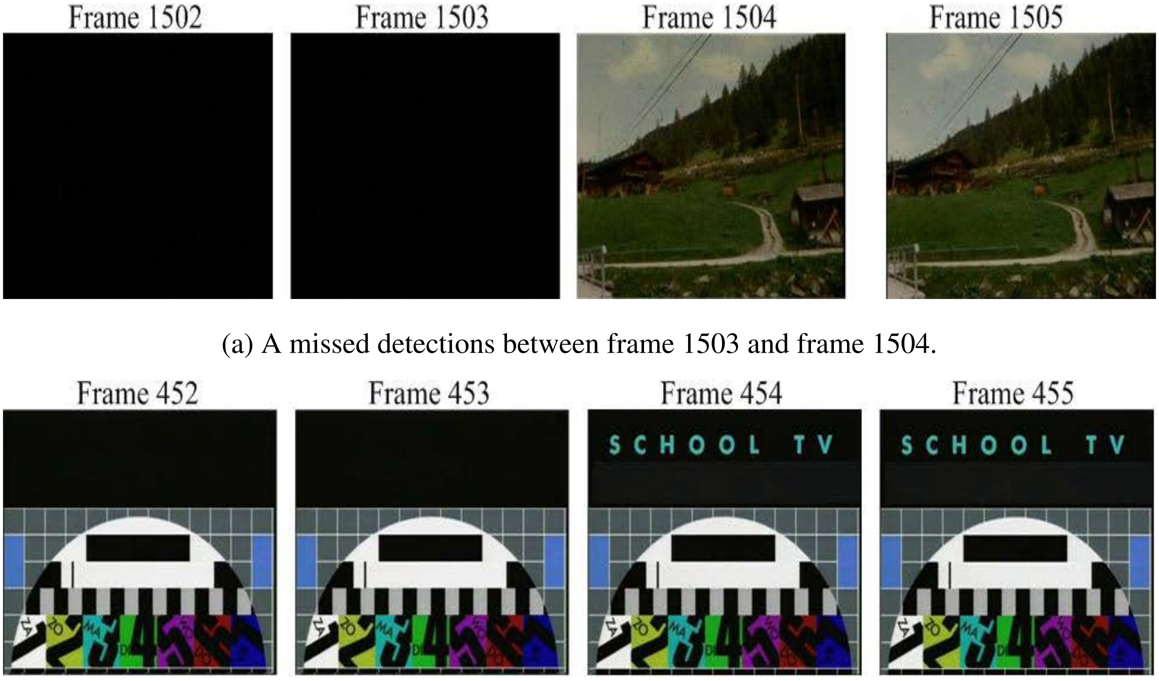
Video name	Proposed Method			Le [63]		
	Recall	Precision	F1	Recall	Precision	F1
BG_2408	0.95	0.98	0.97	0.92	0.95	0.93
BG_37359	0.95	0.93	0.94	0.98	1.00	0.99
BG_35050	0.91	0.99	0.95	0.95	0.92	0.93
BG_36028	0.97	0.97	0.97	0.87	0.79	0.83
BG_37417	0.99	0.97	0.98	0.95	0.74	0.83
BG_35187	0.89	1.00	0.94	0.85	0.91	0.88
BG_36537	0.90	0.94	0.92	0.96	0.99	0.97
BG_37879	0.98	0.98	0.98	0.78	0.99	0.87
Average	**0.95**	**0.97**	**0.96**	0.91	0.91	0.90



(a) A missed detections between frame 11738 and frame 11738.

(b) A missed detections between frame 11903 and frame 11904.

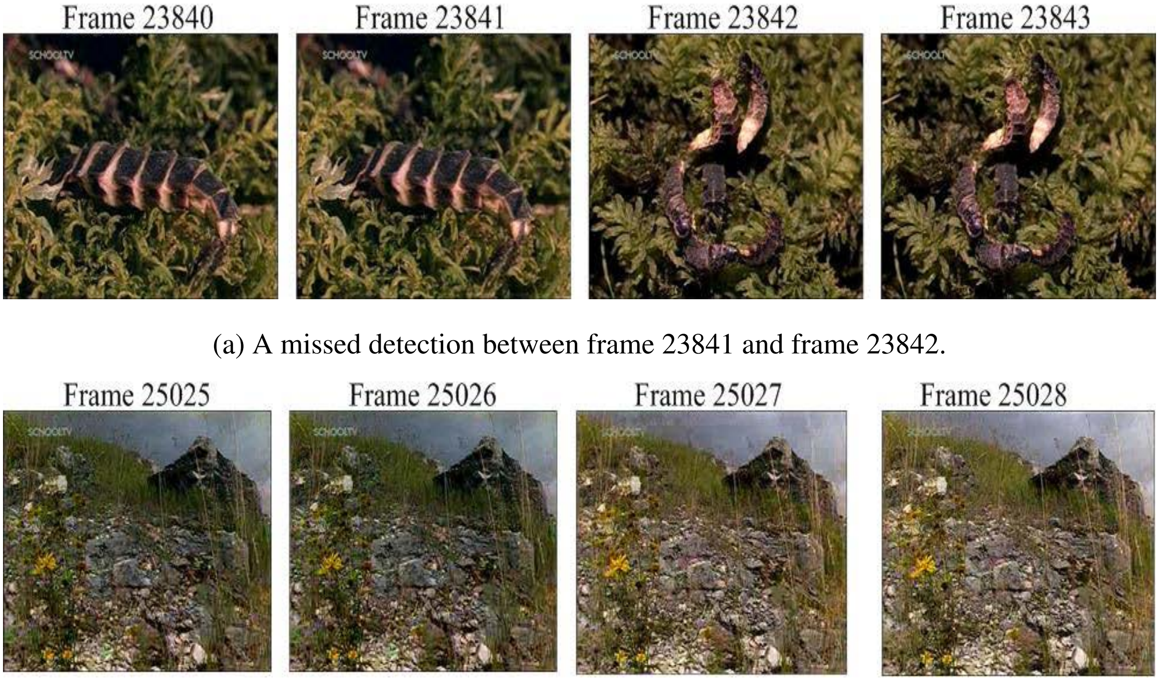
Figure 4.2 Examples of missed detections by the proposed method are taken from video BG_35187.



(a) A missed detections between frame 1503 and frame 1504.

(b) A false detections between frame 453 and frame 454.

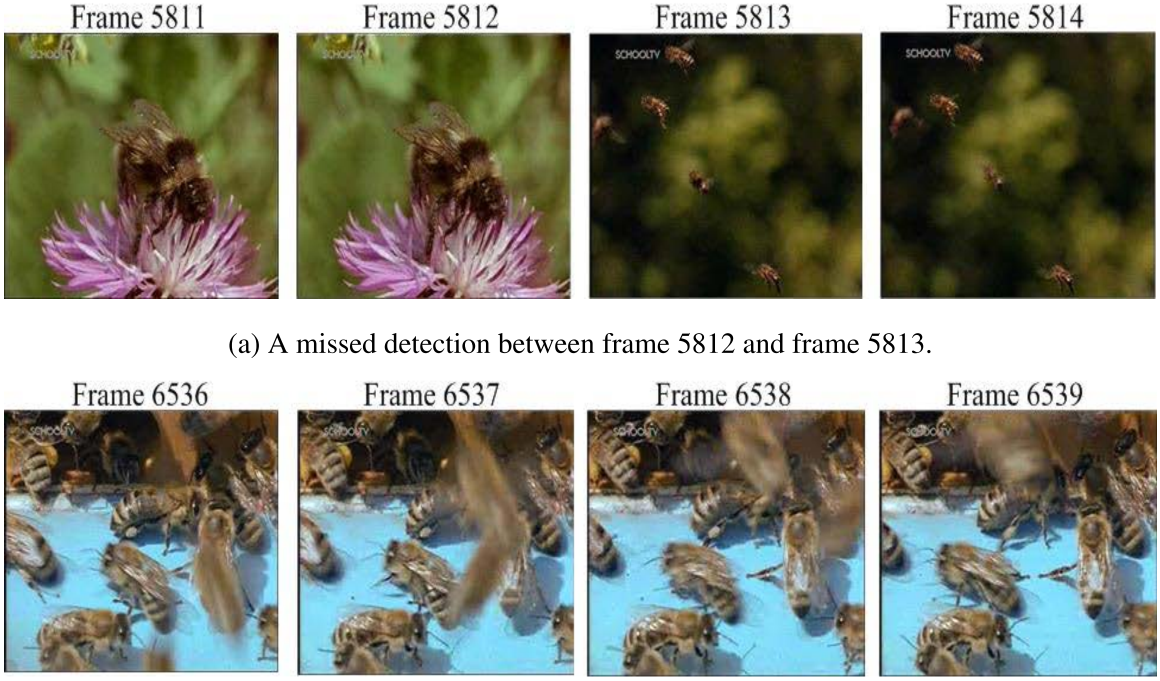
Figure 4.3 Examples of missed and false detection by Le[63] are taken from video BG_35187.



(a) A missed detection between frame 23841 and frame 23842.

(b) A false detection between frame 2506 and frame 2507.

Figure 4.4: Examples of missed and false detections by the proposed method are taken from video BG_37879.



(a) A missed detection between frame 5812 and frame 5813.

(b) A false detection between frame 6537 and frame 6538.

Figure 4.5 Examples of missed and false detections by Le[63] are taken from video BG_37879.

for each frame. Therefore, if the algorithm detected cut between frames f_t and f_{t+1} whose magnitude is larger than a threshold Th_{motion}, these cuts are rejected as motions from large objects. The short shots with less than 25 frames (1 second) are removed. Next, Sub-shot segmentation algorithm in [8] is used to divide shots into smaller units. A first frame of the shot is chosen as the base frame b and next frame c for comparison. The χ^2 distance used to compute the distance of frame sequence until the sum of the sorted value of lower eight is larger than a threshold $Th_{sub-shot}$. The frames from b to $c - 1$, then, form a sub-shot and frame c is used as the next base frame. The short sub-shots with less than 25 frames are removed. Next, We employ keyframe extraction algorithm proposed in [70] to extract the representative keyframes from each sub-shot. In this approach, cosine distance is used to measure the difference between neighboring frames in sub-shot. Keyframes are selected at the midpoints between two consecutive high curvature points where the high curvature points are detected from the curve of the cumulative frame difference.

4.4.1 Color Bars Removal

In this experiment is used to test an accuracy of the proposed method for remove a color bar keyframes. The data sets used in this experiment are obtained from TRECVID 2007 and 2008 rushes summarization data set that consisted of five videos. The videos are in MPEG-1 format with frame rate of 29.97 fps and the frame size of 352×288 pixels. The set of ground truth are manually identified. In order to test an accuracy, the set of keyframes are divided into 4×4 blocks, and then extracted a local RGB histogram with 64-bin per channel. The χ^2 distance is used to compute the histogram differences between any two neighboring blocks in each column by using equation 3.8 and 3.9. Next, these values are sorted into ascending order. If the value of 10^{th} is smaller than the threshold θ_{cb}, then keyframe is defined as a color bar keyframe.

The result is shown in Table 4.6, the propose method provides high recall and precision value on average. Therefore, the proposed has high performance to detect and remove a color bars keyframe. Due to a color bars frame has a unique characteristic and there is appears less than two times in input video. Then, it is easy to detect and remove.

Table 4.6 The result of color bars removal

Video name	**Proposed Method**		
	Recall	**Precision**	**F1**
MRS035126	1.00	1.00	1.00
MRS044499	1.00	1.00	1.00
MRS044725	1.00	1.00	1.00
MRS045104	1.00	1.00	1.00
MRS145332	1.00	1.00	1.00
Average	**1.00**	**1.00**	**1.00**

4.4.2 Single Color Removal

In this experiment is used to test an accuracy of the proposed method for remove a single color keyframe. The data sets used in this experiment are obtained from TRECVID 2007 and 2008 rushes summarization data set that consisted of five videos. The videos are in MPEG-1 format with frame rate of 29.97 fps and the frame size of 352×288 pixels. The set of ground truth are manually identified. In order to test an accuracy, the set of keyframes are divided into 4×4 blocks, and then extracted a local RGB histogram with 64-bin per channel. Next, sum of average of the i^{th} bin of the RGB histogram is determined by using equation 3.10. If the i^{th} of the global color histogram is larger than the threshold θ_{single} then keyfram is defined as a single color keyframe.

The result is shown in Table 4.7, the proposed method provides high recall and precision value.

Therefore, the proposed method has a high performance to detect and remove a single color keyframe. Due to a single color keyframe has a unique characteristic. Then, it is easy to detect and remove. However, video MRS044725 and MRS048773 have a low precision value. Because, some keyframes have a characteristic same as a single color frame as shown in Figure 4.6. Moreover, they were recorded with the low-light condition.

Table 4.7 The result of single color removal

Video name	**Proposed Method**		
	Recall	**Precision**	**F1**
MRS035126	1.00	0.71	0.83
MRS044499	1.00	1.00	1.00
MRS145332	1.00	1.00	1.00
MRS145343	1.00	0.75	0.86
MRS148797	1.00	0.90	0.95
Average	**1.00**	**0.87**	**0.93**

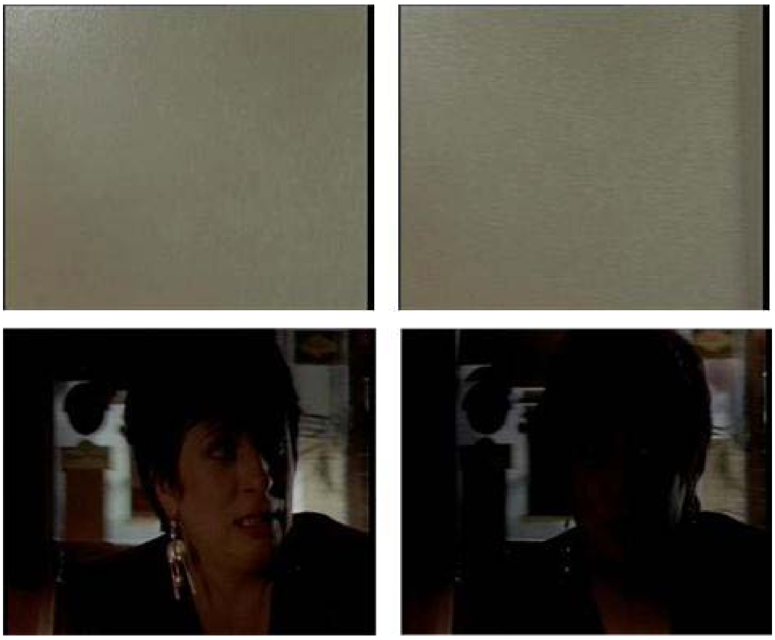


Figure 4.6: Examples of false detections by the proposed method are taken from video MRS035126 and MRS0145343.

4.4.3 Clapper Board Removal

In this experiment is used to test an accuracy of NDK method for remove a clapper board keyframe. The data sets used in this experiment are obtained from TRECVID 2007 and 2008 rushes summarization data set that consisted of five videos. The videos are in MPEG-1 format with frame rate of 29.97 fps and the frame size of 352x288 pixels. The set of ground truth are manually identified. In order to test an accuracy, a set of 80 example frames of the clapper board are extracted from the TRECVID 2007 and 2008 rushes summarization development set. The backgrounds where clapper boards are not present were manually removed. The key points are extracted for each frame, and then they used as a set of queries. The input keyframes are extracted the key points, and match them with the set of queries by using NDK algorithm. If the result of the NDK algorithm returns a matched with a query, then keyframe is defined as a clapper board keyframe.

The result is shown in Table 4.8, the NDK method provides high recall and precision value on average. It implies that the NDK method has a high performance to detect and remove a clapper board frame. However, video MRS148090 has a low recall due to it consists of the large variations of clapper board as shown in Figure 4.7. In additional, the clapper boards are quick move in and move out before recoded a film. Moreover, a method for extract keyframes is based on high curvature points. Then, some high movement of the clapper board keyframes were extracted as a keyframe representative. Thus, the given keyframe has affected with a motion blur. Therefore, the NDK method cannot detect and remove this keyframe.

Table 4.8 The result of clapper boards removal

Video name	**Proposed Method**		
	Recall	**Precision**	**F1**
MRS035126	0.81	1.00	0.89
MRS044499	1.00	1.00	1.00
MRS145343	0.68	1.00	0.81
MRS151099	0.77	1.00	0.87
MRS146570	0.86	1.00	0.92
Average	**0.82**	**1.00**	**0.90**

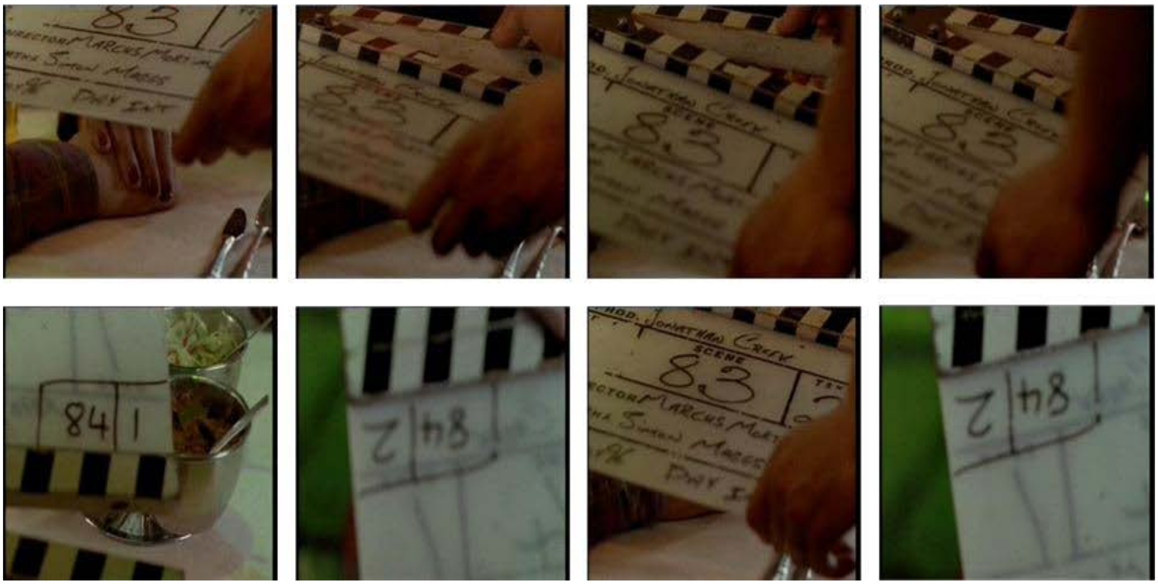


Figure 4.7 Examples of missed detections by the NDK method are taken from video MRS145343.

4.5 Retake Detection

4.5.1 Manually extract the shot boundary

In this experiment is used to compare the performance of the proposed method and the existing method by manually extract the video shot boundary. The method by Bailer [20] provides the best performance of retake detection with the TRECVID 2007 rushes summarization data set. Therefore, to evaluate the retake detection performance, the proposed method is compared with Bailer [20]'s method by setting the parameter as recommended in [20]. The datasets used in this experiment are obtained from TRECVID 2007 and 2008 rushes summarization data set that consisted of five videos. The videos are in MPEG-1 format with frame rate of 29.97 fps and the frame size of 352×288 pixels. The input videos are divided into segment by using manually shot boundary. The set of ground truth are manually identified. For example, video V is divided into six shots. Shots one to four inclusive are a take in the first scene whilst shots five and six are a take in the second scene. Then the ground truth of video V is set equal to two retakes. The experiment setup for the proposed method is first extracted keyframe from each shot at every 10^{th} frame. Each frame is divided into $B \times B$ blocks. From our experiment, $B = 5$ provided the best result as shown in Table 4.9. Each block is extracted the SIFT feature. The similarity between consecutive frames is calculated using SIFT matching, and then converted into a string. The given string is then concatenated into a string sequence to use as the shot representative. The similarity between two sequences is evaluated by the LCS algorithm. The algorithm one is performed to detect the retake. The performance comparison is shown in Table 4.9.

Video MRS151585 has perfect matching results, which reflects the fact that in this video only few objects (Actors) appeared in a scent, and that in addition the motion magnitude in video was also low which increases the SIFT feature matching result as shown in Figure 4.8. On the other hand, Bailer [20] has low recall and precision due to his algorithm creates the take candidate by using pairwise matching of shots. Then, takes that have the same visual content were merging together. Therefore, his algorithm cannot detect them. Video MRS150072 gave a low recall with this proposed method, probably because the retakes in the same scene have a different duration and so contain different amounts of information between takes within the same scene as shown in Figure 4.9. In this scenario, the differences in the duration of each take will produce a different length string and, therefore, the number of matched string by the LCS algorithm will be less. Therefore, from this experimental result indicated that the proposed method has a better performance to detect a retake than Bailer [20].

Table 4.9 Performance comparison with manual shot boundary detection.

Video name	**Proposed Method**			**Bailer [20]**		
	Recall	**Precision**	**F1**	**Recall**	**Precision**	**F1**
MRS151585	1.00	1.00	1.00	0.83	0.83	0.83
MRS150072	0.78	1.00	0.88	0.89	1.00	0.94
MRS025913	0.86	1.00	0.92	0.63	0.63	0.63
MRS044500	0.80	1.00	0.89	0.80	1.00	0.89
MRS145918	0.83	1.00	0.91	0.67	1.00	0.80
Average	**0.85**	**1.00**	**0.92**	0.76	0.89	0.82

4.5.2 Automatic extract the shot boundary

In this experiment is used to compare a performance of the proposed method and existing method by automatic extract the video shot boundary. The method by Bailer [20] provides the best performance in retake detection with the TRECVID 2007 rushes summarization data set. Therefore, to evaluate the retake detection performance, the proposed method is compared with Bailer [20]'s method by setting the parameter as recommended in [20]. The data sets used in this experiment are obtained from TRECVID 2007 and 2008 rushes summarization data set that consisted of five videos.

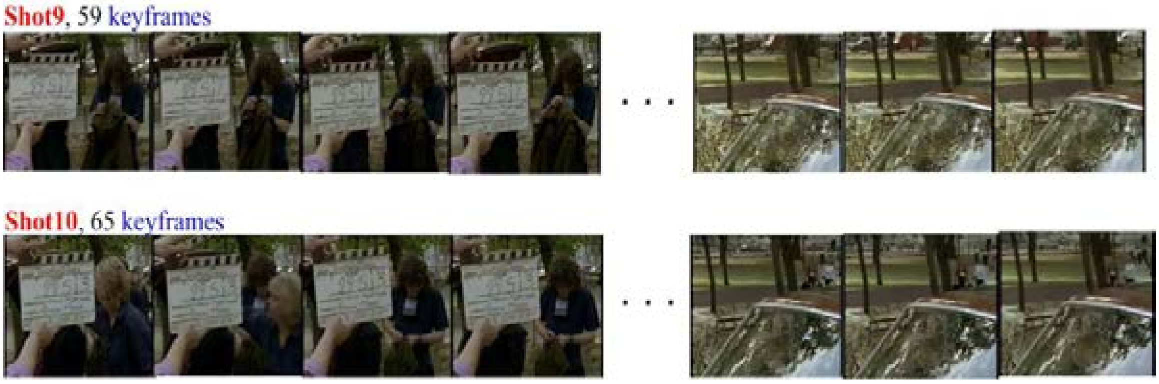


Figure 4.8 Examples of retake that is taken from video MRS151585.

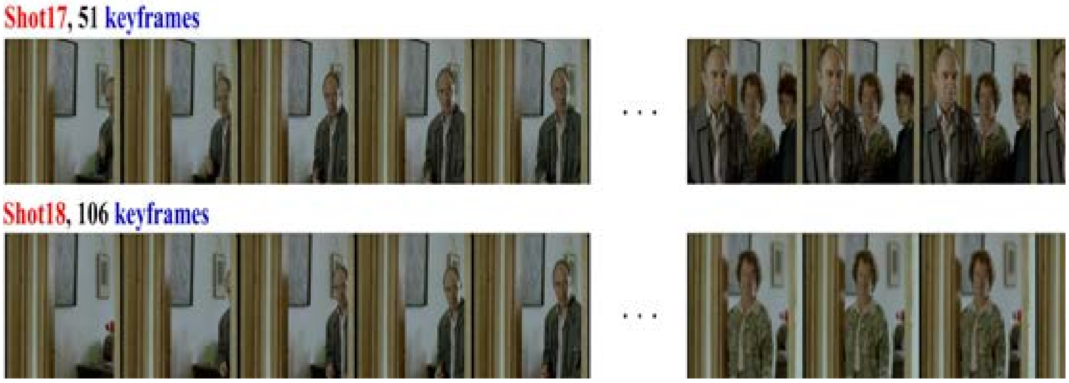


Figure 4.9 Examples of retake that is taken from video MRS150072.

The videos are in MPEG-1 format with frame rate of 29.97 fps and the frame size of 352×288 pixels. The input videos are divided into segment by using the proposed automatic shot boundary. The set of ground truth from previous experiment was used for evaluate the results. The experiment setup for the proposed method is first extracted keyframe from each shot at every 10^{th} frame. Each keyframe is divided into $B \times B$ blocks. From our experiment, $B = 5$ provided the best result as shown in the previous experiment. Each block is extracted the SIFT features. The similarity between consecutive frames is calculated using SIFT matching, and then converted into a string. The given string is then concatenated into a string sequence to use as shot representative. The similarity between two sequences is evaluated by using the LCS algorithm. The algorithm one is performed to detected the retake.The performance comparison is shown in Table 4.10. For video MRS15185, MRS150072, MRS044500 and MRS145918, the proposed method and Bailer [20] have the results same as the previous experiment, due to the results of shot boundary that extracted by using the proposed method are closely to the ground truth. Video MRS044500 gave a low accuracy when compare with the previous experiment, due to the result of shot boundary that extracted by the proposed method are different from the ground truth. Then, some retakes are cannot detect. It implies that, the performance of retake detection depends on the performance of shot boundary detection. However, From the average recall, precision and F1 indicated that the proposed method has a better performance to detect a retake than Bailer [20].

Table 4.10 Performance comparison with automatic shot boundary detection.

Video name	**Proposed Method**			**Bailer [20]**		
	Recall	**Precision**	**F1**	**Recall**	**Precision**	**F1**
MRS151585	1.00	1.00	1.00	0.83	0.83	0.83
MRS150072	0.78	1.00	0.88	0.89	1.00	0.94
MRS025913	0.63	1.00	0.77	0.38	0.50	0.43
MRS044500	0.80	1.00	0.89	0.80	1.00	0.89
MRS145918	0.83	1.00	0.91	0.67	1.00	0.80
Average	**0.81**	**1.00**	**0.89**	0.71	0.87	0.78

CHAPTER V

CONCLUSION

5.1 Conclusion

In this dissertation, a new approach to detect the presence of a retake in rushes video based on matching a string sequence encoded from the location of the object was presented. The object recognition, based on the SIFT features, is used to extract the location of the object. The framework of this proposed method was designed into four steps. First, the input video is decomposed into shots by using a local Singular Value Decomposition (SVD) and k-means clustering. Second, useless shots such as color bars, single colors, very short shots and clapper boards are removed. Third, keyframes are then extracted from each shot at every 10^{th} frame. The given frames are encoded into a string sequence using the location of the object based on the SIFT features. Finally, the LCS and simple algorithm are, then, enacted to detect the presence of any retakes. From the experimental results, the proposed method has a better performance to detect the presence of any retakes than the existing retake detection algorithms. The main contributions of the proposed method can be summarized as follows:

- The automatic shot boundary detection was proposed to organize the video data into segments. Form the experimental results, the proposed method provides a better performance than the other method. However, the proposed method cannot detect some adjacent shots that have similar backgrounds and visual contents which must be improved.
- The automatic retake detection is able to detect retake which included in rushes video. The proposed method was found to work well with videos that contained a low motion magnitude and few objects. The accuracy of the proposed method is depended on the performance of the automatic shot boundary detection. From the performance comparison, the proposed method provides a better performance than the existing methods.

5.2 Future Work

Integrating the motion information in order to solve the problem and produce the performance result of retake detection should be future investigated. Moreover, the PCA-SIFT algorithm can be used to improve the performance of the proposed method.

REFERENCES

[1] Over, P., Smeaton, A. F., and Kelly, P., The TRECVID 2007 BBC Rushes Summarization Evaluation Pilot. in Proceedings of the international workshop on TRECVID video summarization(TVS '07), New York, NY, USA, ACM, (2007): 1–15.

[2] Over, P., Smeaton, A. F., and Awad, G., The TRECVid 2008 BCC Rushes Summarization Evaluation. in Proceedings of the 2nd ACM TRECVid Video Summarization Workshop(TVS '08), New York, NY, USA, ACM, (2008): 1–20.

[3] Katiyar, A. and Weissman, J., ViDeDup: An Application-Aware Framework for Video Deduplication. in Proceedings of HotStorage 2011: 3rd Workshop on Hot Topics in Storage and File Systems.

[4] Lijin, Z. Analysis and Detection of Redundant Data in Sports Videos - Take Trampoline Videos as the Example. International Journal of Advancements in Computing Technology 1 (2011): 161-169.

[5] Gao, Y., Wang, W., and Yong, J. A Video Summarization Tool using Two-Level Redundancy Detection for Personal Video Recorders. IEEE Transactions on Consumer Electronics 54 (2008): 521-526.

[6] Li, Y., Jin, J., and Zhou, X., Matching Commercial Clips from TV Streams Using a Unique, Robust and Compact Signature. in Proceedings of the Digital Image Computing: Techniques and Applications(DICTA 2005), Washington, DC, USA, IEEE Computer Society, (2005): 355–362.

[7] Li, Y., Jin, J., and Zhou, X., Matching Commercial Clips from TV Streams Using a Unique, Robust and Compact Signature. in Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communications Systems (ISPAC 2005), Washington, DC, USA, IEEE Computer Society, (2005): 317–320.

[8] Pan, C.-M., Chuang, Y.-Y., and Hsu, W. H., NTU TRECVID-2007 Fast Rushes Summarization System. in Proceedings of the international workshop on TRECVID video summarization(TVS '07), New York, NY, USA, ACM, (2007): 74–78.

[9] Liu, Y., Liu, Y., and Zhang, Y., The Hong Kong Polytechnic University at TRECVID 2007 BBC Rushes Summarization. in Proceedings of the international workshop on TRECVID video summarization(TVS '07), New York, NY, USA, ACM, (2007): 50–54.

[10] Truong, B. T. and Venkatesh, S., Generating Comprehensible Summaries of Rushes Sequences based on Robust Feature Matching. in Proceedings of the international workshop on TRECVID video summarization(TVS '07), New York, NY, USA, ACM, (2007): 30–34.

[11] Wang, F. and Ngo, C.-W., Rushes Video Summarization by Object and Event Understanding. in Proceedings of the international workshop on TRECVID video summarization(TVS '07), New York, NY, USA, ACM, (2007): 25–29.

[12] Toharia, P., Robles, O., Pastor, L., and Rodriguez, A., Combining Activity and Temporal Cherence with Low-level Information for Summarization of Video Rushes. in Proceedings of the international workshop on TRECVID video summarization(TVS '08), New York, NY, USA, ACM, (2008): 70–74.

[13] Chasanis, V., Likas, A., and Galatsanos, N., Video Rushes Summarization Using Spectral Clustering and Sequence Alignment. in Proceedings of the international workshop on TRECVID video summarization(TVS '08), New York, NY, USA, ACM, (2008): 75–79.

[14] Liu, Z., Zavesky, E., Shahraray, B., Gibbon, D., and Basso, A., Brief and High-Interest Video Summary Generation: Evaluating the AT&T Labs Rushes Summarizations. in Proceedings of the international workshop on TRECVID video summarization(TVS '08), New York, NY, USA, ACM, (2008): 21–25.

[15] Bai, L., Lao, S., Smeaton, A., and O'Connor, N., Automatic Summarization of rushes video using bipartite graphs. in Proceedings of 3rd International Conference on Semantic and Multimedia Technologies, Berlin, Heidelberg, Springer-Verlag, (2008): 3–14.

[16] Bai, L., Hu, Y., Lao, S., Smeaton, A., and O'Connor, N. Automatic Summarization of rushes video using bipartite graphs. Multimedia Tools Application 49 (2009): 63-80.

[17] Ren, J. and Jiang, J. Hierarchical Modeling and Adaptive Clustering for Real-Time Summarization of Rush Videos. IEEE Transaction on Multimedia 11 (2009): 906-917.

[18] Emilie, D. and Bernard, M., Sequence Alignment for Redundancy Removal in Video Rushes Summarization. in Proceedings of the international workshop on TRECVID video summarization(TVS '08), New York, NY, USA, ACM, (2008): 55–59.

[19] Emilie, D. and Bernard, M. Rushes Video Summarization and Evaluation. Journal Multimedia Tools and Applications 48 (2010): 63-80.

[20] Bailer, W., Lee, F., and Thallinger, G. A Distance Measure for Repeated Takes of One Scene. The Visual Computer 25 (2009): 53-68.

[21] Bailer, P., A Comparison of Distance Measures for Clustering Video Sequences. in Proceedings of 19th International Conference on Databased and Expert Systems Application(DEXA '08), Washington, DC, USA, IEEE Computer Society, (2008): 595–599.

[22] Smeaton, A. F., Kraaij, W., and Over, P., TRECVID 2003 - An Overview. in Proceedings of TRECVID 2003 - Text REtrieval Conference TRECVID Workshop, Gaithersburg, MD, USA, NIST.

[23] Kraaij, W., Smeaton, A. F., Over, P., and Arlandis, J., TRECVID 2004 - An Overview. in Proceedings of TRECVID 2004 - Text REtrieval Conference TRECVID Workshop, Gaithersburg, MD, USA, NIST.

[24] Over, P., Ianeva, T., Kraaijz, W., and Smeaton, A. F., TRECVID 2005 - An Overview. in Proceedings of TRECVID 2005 - Text REtrieval Conference TRECVID Workshop, Gaithersburg, MD, USA, NIST.

[25] Over, P., Ianeva, T., Kraaijz, W., and Smeaton, A. F., TRECVID 2006 - An Overview. in Proceedings of TRECVID 2006 - Text REtrieval Conference TRECVID Workshop, Gaithersburg, MD, USA, NIST.

[26] Over, P., Awad, G., Kraaij, W., and Smeaton, A. F., TRECVID 2007 - An Overview. in Proceedings of TRECVID 2007 - Text REtrieval Conference TRECVID Workshop, Gaithersburg, MD, USA, NIST.

[27] Over, P., Awad, G., Rose, T., Fiscus, J., Kraaij, W., and Smeaton, A. F., TRECVID 2008 Goals, Tasks, Data, Evaluation Mechanisms and Metrics. in Proceedings of TRECVID 2008 - Text REtrieval Conference TRECVID Workshop, Gaithersburg, MD, USA, NIST.

[28] Over, P., Awad, G., Fiscus, J., Michel, M., Smeaton, A., and Kraaij, W., TRECVID 2009 Goals, Tasks, Data, Evaluation Mechanisms and Metrics. in Proceedings of TRECVID 2009 - Text REtrieval Conference TRECVID Workshop, Gaithersburg, MD, USA, NIST.

[29] Over, P., Awad, G., Fiscus, J., Antonishek, B., Michel, M., Smeaton, A., Kraaij, W., and Quenot, G., TRECVID 2010 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms, and Metrics. in Proceedings of TRECVID 2010 - Text REtrieval Conference TRECVID Workshop, Gaithersburg, MD, USA, NIST.

[30] Smeaton, A. F., Over, P., and Kraaij, W., Evaluation Campaigns and TRECVid. in Proceedings of the 8th ACM international workshop on Multimedia information retrieval(MIR '06), New York, NY, USA, ACM, (2006): 321–330.

[31] Snoek, C. G. M. and Worring, M. Multimodal Video Indexing: A Review of the State-of-the-art. Multimedia Tools and Applications 25 (2005): 5-35.

[32] Putpuek, N., Cooharojananone, N., Lursinsap, C., and Satoh, S., Unified Approach to Detection and Identification of Commercial Films by Temporal Occurrence Pattern. in Proceedings of 20th International Conference on Pattern Recognition(ICPR '10), Washington, DC, USA, IEEE Computer Society, (2010): 3288–3291.

[33] Xiong, Z., Radhakrishnan, R., Divakaran, A., Rui, Y., and Huang, T. S. A Unified Framework for Video Summarization, Browsing & Retrieval: with Applications to Consumer and Surveillance Video. Academic Press, 2005.

[34] Cotsaces, C., Nikolaidis, N., and Pitas, I. Video Shot Boundary Detection and Condensed Representation: A Review. IEEE Signal Processing Magazine 23 (2006): 28-37.

[35] Manickam, N., Parnami, A., and Chandran, S., Reducing False Positives in Video Shot Detection Using Learning Techniques. in Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing(ICVGIP 2006), Berlin, Heidelberg, Springer-Verlag, (2006): 421–432.

[36] Smith, M. A. and Kanade, T. Multimodal Video Characterization And Summarization. Springer, 2004.

[37] Furht, B. Encyclopedia of Multimedia. Springer, 2nd ed., 2005.

[38] Golub, G. and Kahan, W. Calculating the Singular Values and Rseudo-Inverse of a Matrix. Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis 2 (1965): 205-224.

[39] Golub, G. and Reinsch, C. Singular Value Decomposition and Least Squares Solutions. Numerische Mathematik 14 (1970): 403-420.

[40] Stewart, G. On the Early History of the Singular Value Decomposition. SIAM Review 35 (1993): 551-566.

[41] Andrews, H. and Patterson, C. Singular Value Decompositions and Digital Image Processing. IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-24 (1976): 26-53.

[42] Jeong, J. and Ha, Y., Video Sequence Matching Using Singular Value Decompositions. in Proceedings of International Conference on Image Analysis and Recognition(ICIAR '2006), Berlin, Heidelberg, Springer-Verlag, (2006): 426–435.

[43] Cernekov, Z., Kotropoulos, C., and Pitas, I. Video shot-boundary detection using singular-value decomposition and statistical test. Journal of Electronic Imaging 16 (2007): 51-59.

[44] Forgy, E. W. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Biometrics 21 (1965): 768-780.

[45] MacQueen, J., Some Methods for classification and Analysis of Multivariate Observations. in Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, California, USA, Berkeley, University of California Press, (1967): 281–297.

[46] Lloyd, S. Least Squares Quantization in pcm. IEEE Transactions on Information Theory 28 (1982): 129-136.

[47] Jain, A., Murty, M. N., and Flynn, P. J. Data clustering: A review. Journal ACM Computing Surveys (CSUR) 31 (1999): 264-323.

[48] Lowe, D. G., Object recognition from local scale-invariant features. in Proceedings of the Seventh IEEE International Conference on Computer Vision(ICCV '99), Washington, DC, USA, IEEE Computer Society, (1999): 1150–1157.

[49] Lowe, D. G., Local feature view clustering for 3D object recognition. in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR 2001), Washington, DC, USA, IEEE Computer Society, (2001): 682–688.

[50] Lowe, D. G. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60 (2004): 91-110.

[51] Mikolajczyk, K. and Schmid, C. A Performance Evaluation of Local Descriptors. Journal IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005): 1615-1630.

[52] Schugerl, P., Sorschag, R., Bailer, W., and Thallinger, G., Object Re-detecting Using SIFT and MPEG-7 Color Descriptors. in Proceedings of the 2007 international conference on Multimedia content analysis and mining(MCAM'07), Berlin, Heidelberg, Springer-Verlag, (2007): 305–314.

[53] Ruiz-Del-Solar, J., Loncomilla, P., and Zorzi, P., Applying SIFT Descriptors to Stellar Image Matching. in Proceedings of the 13th Iberoamerican congress on Pattern Recognition: Progress in Pattern Recognition, Image Analysis and Applications(CIARP'08), Berlin, Heidelberg, Springer-Verlag, (2008): 618–625.

[54] Xu, D., Cham, T.-J., Yan, S., and Chang, S.-F., Near Duplicate Image Identification with Spatially Aligned Pyramid Matching. in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR'08), Washington, DC, USA, IEEE Computer Society, (2008): 1–7.

[55] Fan, Q., Barnard, K., Amir, A., Efrat, A., and Lin, M., Matching slides to presentation videos using SIFT and scene background matching. in Proceedings of the 8th ACM international workshop on Multimedia information retrieval(MIR'06), New York, NY, USA, ACM, (2006): 239–247.

[56] Ngo, C.-W., Zhao, W.-L., and Jiang, Y.-G., Fast tracking of near-duplicate keyframes in broadcast domain with transitivity propagation. in Proceedings of the 14th annual ACM international conference on Multimedia(MULTIMEDIA '06), New York, NY, USA, ACM, (2006): 845–854.

[57] Zhao, W., Ngo, C., Tan, H., and Wu, X. Near-Duplicate Keyfeame Identification With Interest Point Matching and Pattern Learning. IEEE Transactions on Multimedia 9 (2007): 51-59.

[58] Smeaton, A., Over, P., and Doherty, A. Video Shot Boundary Detection: Seven Years of TRECVid Activity. Journal Computer Vision and Image Understanding 114 (2010): 411-418.

[59] Boreczky, J. and Rowe, L. Comparison of video shot boundary detection. Journal of Electronic Imaging 5 (1996): 122-128.

[60] Suzuki, K., Nakajima, M., Sakano, H., Sambe, Y., and Ohtsuka, S., Abrupt Shot Boundary Detection from Video Sequence Using Motion Direction Histogram Feature. in Proceedings of IAPR Workshop on Machine Vision Applications, (2002): 572–575.

[61] Cernekova, Z., Kotropoulos, C., and Pitas, I., Video Shot Segmentation using Singular Value Decomposition. in Proceedings of the 2003 International Conference on Multimedia and Expo(ICME '03), Washington, DC, USA, IEEE Computer Society, (2003): 181–184.

[62] Cernekova, Z., Kotropoulos, C., Nikolaidis, N., and Pitas, I., Video Shot Segmentation using Fusion of SVD and Mutual Information Features. in Proceedings of 2005 IEEE International Symposium on Circuits and Systems, Washington, DC, USA, IEEE Computer Society, (2005): 3849–3852.

[63] Le, D., Satoh, S., Ngo, T., , and Duong, D., A Text Segmentation Based Approach to Video Shot Boundary Detection. in Proceedings of 2008 IEEE 10th Workshop on Multimedia Signal Processing, Washington, DC, USA, IEEE Computer Society, (2008): 702–706.

[64] Le, D.-D. and Satoh, S., National Institute of Informatics, Japan at TRECVID 2007: BBC Rushes Summarization. in Proceedings of the international workshop on TRECVID video summarization(TVS '07), New York, NY, USA, ACM, (2007): 70–73.

[65] Ren, J., Jiang, J., and Eckes, C., Hierarchical Modeling and Adaptive Clustering for Realtime Summarization of Rush Videos in TRECVID08. in Proceedings of the international workshop on TRECVID video summarization(TVS '08), New York, NY, USA, ACM, (2008): 26–30.

[66] Noguchi, A. and Yanai, K., Rushes Summarization Based on Color, Motion and Face. in Proceedings of the international workshop on TRECVID video summarization(TVS '08), New York, NY, USA, ACM, (2008): 139–143.

[67] Van Siclen, C. D. Information measure of location precision. Journal Applied Mathematics and Computation 97 (1998): 287-293.

[68] Sau, B. and Mukhopadhyaya, K., Locating Objects in a Sensor Grids. in Proceedings of 6th International Workshop on Distributed Computing(IWDC'04), Washington, DC, USA, IEEE Computer Society, (2004): 526–531.

[69] Putpuek, N., Le, D.-D., Cooharojananone, N., Satoh, S., and Lursinsap, C., Rushes Summarization Using Different Redundancy Elimination Approaches. in Proceedings of the international

workshop on TRECVID video summarization(TVS '08), New York, NY, USA, ACM, (2008): 100–104.

[70] Gaianluigi, C. and Raimondo, S. An innovative algorithm for key frame extraction in video summarization. Journal of Real-Time Image Processing 1 (2006): 69-88.

Appendix

Appendix
List of Publications

- Xiaomeng Wu, **Narongsak Putpuek**, and Shin'ichi Satoh, "Commercial Film Detection and Identification Based on a Dual-Stage Temporal Recurrence Hashing Algorithm", in *Proc. of Workshop on Very-Large-Scale Multimedia Corpus, Mining and Retrieval (VLSMCMR2010), in conjunction with ACM Multimedia*, Firenze, Italy, October 29, pp. 3288-3291, 2010.

- **Narongsak Putpuek**, Nagul Cooharojananone, Chidchanok Lursinsap and Shin'ichi Satoh, "Unified Approach to Detection and Identification of Commercial Films by Temporal Occurrence Pattern", in *Proc. of the* 20^{th} *International Conference in Pattern Recognition (ICPR2010)*, Istanbul, Turkey, August 23-26, pp. 3288-3291, 2010.

- Nagul Cooharojananone, **Narongsak Putpuek**, Shin'ichi Satoh, and Chidchanok Lurinsap, "A Novel Retake Detection using LCS and SIFT Algorithm", in *Proc. of IEEE Pacific-Rim Conference on Multimedia 2009 (PCM 2009)*, Bangkok, Thailand, December 15-18, pp. 777-787, 2009.

- **Narongsak Putpuek**, Duy-Dinh Le, Nagul Cooharojananone, Shin'ichi Satoh, Chidchanok Lurinsap, "Rushes Summarization Using Different Redundancy Elimination Approaches", in *Proc. of ACM Multimedia TRECVID BBC Rushes Summarization Workshop (TVS 2008)*, Vancouver, Canada, October 27 - November 1, pp. 100-104, 2008.

- **Narongsak Putpuek**, Nagul Cooharojananone, and Chidchanok Lursinsap, "Complex Human Motions Analysis using an Adaptive Star Skeleton", in *Proc. of the 2nd International Conference on Advances in Information Technology (IAIT 2007)*, Bangkok, Thailand, November 1-2, pp. 187-192, 2007.

Biography

Name: Mr Narongsak Putpuek

Date of Birth: 15^{th} March, 1975

Education:

- Ph.D. Program in Computer Science, Chulalongkorn University, Thailand (October 2006 - September 2011).
- Internship Student, National Institute of Informatics (NII), Japan (December 2007 - May 2008).
- Internship Student, National Institute of Informatics (NII), Japan (November 2009 - March 2010).
- M.Sc. Program in Information Technology(Information Science), King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand (June 2001 - September 2005).
- B.Sc. Program in Electronic and Computer, King Mongkut's Institute of Technology Ladkrabang (KMITL), (June 1996 - March 1998).

Work: Lecturer, Informaction Technology Department, Faculty of Science and Technology, Rajabhat Rajanagarindra University.

Scholarship: Grants for Human Resource Development, Rajabhat Rajanagarindra University, Thailand.