# *FEASIBLITY STUDY OF NEAR-DUPLICATE VIDEO RETRIEVAL BASED ON CLUSTERING TECHNIQUES*

**A thesis presented**

**By**

**Yandan Wang**

**To**

**The faculty of Information Technology**

**in fulfillment of the requirements**
**forthe degree of**
**Doctor of Philosophy**
**in the subject of IT**

**Monash University**
**October, 2013**

# Copyright Notices

## Notice 1

Under the Copyright Act 1968, this thesis must be used only under the normal conditions of scholarly fair dealing. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis

## Notice 2

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

# ABSTRACT

This thesis studies the feasibility of clustering-based Near-Duplicate Video Retrieval (NDVR), which makes use of clustering techniques to pre-process video dataset into Near Duplicate Video (NDV) groups and conducts NDVR on the representatives of clusters. Content based NDVR has been explored for decades. Traditionally, researchers improve the NDVR performance in terms of accuracy and speed through: i) the video feature representation; ii) matching approach; iii) indexing structure.

The proposed clustering-based NDVR approach could increase retrieval speed on one hand and achieves equivalent or even better accuracy compared to non-clustering based NDVR on the other hand. The difference of proposed clustering-based approach from the traditional non-clustering based NDVR is that the unsupervised clustering techniques are considered as the prior dataset process step offline. By such a process, the dataset is well organized into corresponding NDV clusters. It then selects only one video or use cluster centroid (mean vector) to represent the cluster. Instead of comparing the query video to all videos in data collections, it only has to compare to the cluster representatives. All the videos in the cluster will be retrieved when the query video that is compared to the representatives meet the specified threshold.

Theoretically, it is impossible to know that the performance of clustering-based NDVR in terms of retrieval accuracy compared to that of non-clustering based approach. Accordingly, this thesis evaluates the performance of clustering-based NDVR compared to that of non-clustering based under the same criteria.

The evaluation starts with analyzing the clustering algorithms in literature with illustrations as well as the experimental study on what the impact of incorporating these clustering algorithms to pre-cluster the dataset offline for NDVR. After that, a novel clustering framework based on multiple sequence alignment (MSA) is proposed to process video dataset into NDV clusters, and NDVR is conducted on formed clusters. Compared to the other clustering algorithms in literature, the proposed method caters to the variable length of video sequence representation. Finally, it evaluates the MSA clustering-based NDVR by using ordinal, global, and local features respectively. The empirical results show that incorporating clustering algorithms for enhancing NDVR is promising and feasible.

*To My Father and Mother*

# DECLARATION

I hereby declare that this PHD work is composed by me, except where I stated otherwise and cited as appropriate and the work has not been submitted for the award of any degree or academic qualification before.

Yandan Wang

# ACKNOWLEDGEMENTS

This project was done with support of many people. I would like to express my sincerely gratitude toMonash University sunway campus that provide me with funding to support my PHD study.Many thanks to A.Prof. Mohammed Belkhatir who was my lecturer in my Bachelor, gave me encouragement, guidance and assistance from the application of my MPhil till end of my PHD. Also I have to thank my former supervisor Dr.Thomas O'Daniel and Dr.SaadatM.Alhashmi for their support. Meanwhile, I have to thank my current supervisor Prof. Chris Messom and Prof. Guojun Lu who gave me encouragement and supervision.

Lastly, I have to thank Philip Chan who assisted me in how to use the Monash Sun Grid for my some experiments and Hong Kong City University that provide the experiment dataset for free access and download.

# PREFACE

This thesis mainly studies the feasibility of clustering-based near-duplicate video retrieval. Rather than reducing the searching space through indexing structures online, the clustering algorithm is utilized to assist in reducing the searching space offline.

The purpose of this study is to encourage researchers to consider pre-process dataset into near duplicate video clusters before the regular processing, which could benefit the retrieval performance in terms of accuracy and speed.

*Publication list:*

[1] Y. Wang, M. Belkhatir, and B. Tahayna, "Near-duplicate video retrieval based on clustering by multiple sequence alignment," in *Proceedings of the 20th ACM international conference on Multimedia*, 2012, pp. 941–944.

[2] Y. Wang, G. Lu, M. Belkhatir, and C. Messom, "The impact of global and local features on multiple sequence alignment clustering-based near-duplicate video retrieval," Accepted by the 2013 pacific-rim conference on multimedia (PCM 2013).

[3] Y. Wang, and M. Belkhatir, "Studying the impact of clustering on near-duplicate video retrieval," Accepted by International Journal of Multimedia Information Retrieval (IJMIR), 2013.

# *TABLE OF CONTENTS*

# *LIST OF FIGURES*

# *LIST OF TABLES*

# CHAPTER 1

# *INTRODUCTION*

## 1.1. Introduction of the Problem

This section will introduce the problem through two aspects: i) what is near-duplicate video; ii) why is Near Duplicate Video Detection (NDVD)/Near Duplicate Video Retrieval (NDVR) important. In the literature, there is no discrimination between NDVD and NDVR. However, from my perspective, NDVD emphasizes the detection accuracy and is more connected to detecting the copyright issues and commercial monitoring, while NDVR emphasizes the scalability for search purpose. I will define NDVR as an extension of the NDVD. In literature, NDVR is mentioned when it refers to the retrieval in large-scale dataset [97] [98] [99]. In some cases, it can address NDVR as NDVD, for example retrieve NDV from web video collections. However, in some cases, it does not work. For example commercial video monitoring, which is called NDVD, and not proper to be addressed as retrieval. However, NDVR is using NDVD techniques as well. When it refers to large-scale NDVR, indexing structures have to be introduced to speed up the retrieval operation, while in traditional NDVD operations, indexing structures are not introduced. Section 1.1.2 shows why NDVD/NDVR is important.

### 1.1.1. What is Near-Duplicate Video?

Near-duplicate videos are videos that have similar or identical frame content in visual, but they are likely to appear differently due to diverse changes introducedduring:i) video capturing e.g. different camera view angle,set camera property differently, under different lighting condition, different background/foreground view, etc.;ii) transformations e.g. using different frame format, scaling, rotation, frame rate, resolution, shifting, contrasting, brightness, saturation, cropping, blurring, etc.; iii) editing operations e.g. adding frames, borders insertion to frames, frames dropping, caption & logos insertion, re-ordering frames, content modification etc..H.Min et. Al [105] compares different definitions of near duplicate video clip (NDVC) in their related work as following:

*Basic NDVC definitions:*

- "The term copy is employed here for a document obtained from an original by the application of one or several transformations such as filtering, cropping, scaling, insertion of logos or frames, addition of noise, etc. There is significant diversity in the nature and amplitude of the encountered transformation. If the copy is to have some interest, these transformations must nevertheless preserve the main information conveyed by the original content. [111]"
- "Identical or approximately identical videos close to the exact duplicate of each other, but different in file formats, encoding parameters, photometric variations (color, lighting changes), editing operations (caption, logo, and border insertion), different lengths, and certain modifications (frames add/remove). [106]"

*Extended NDVC definitions:*

- "The definition of a near-duplicate image varies depending on what photometric and geometric variations are deemed acceptable. This depends on the application. In the case of exact duplicate detection, no changes are allowed. At the other extreme, a more general definition is to require that images be of the same scene, but with possibly different viewpoints and illumination. [107]"

- "Videos of the same scene (e.g., a person riding a bike), varying viewpoints, sizes, appearances, and camera motions. The same semantic concept can occur under different illumination, appearance, and scene settings, just to name a few. [108]"

- "NDVCs are approximately identical videos that might differ in encoding parameters, photometric variations (color, lighting changes), editing operations (captions, or logo insertion), or audio overlays. Conversely, identical videos with relevant complementary information in any of them (changing clip length or scenes) are not considered as NDVCs. Furthermore, users perceive as near-duplicates videos that are not alike but that are visually similar and semantically related. In these videos, the same semantic concept must occur without relevant additional information (i.e., the same information is presented under different scene settings). [109]"

### 1.1.2. Why is Content-based Near-Duplicate Video Detection/Retrieval Important?

#### *Commercial Video Monitoring*

The TV end users are always interested to record some TV programs of interest for the collection or material references purpose. However, due to the insertion of some commercial advertisements and the exact broadcasting time is not known, some

undesirable commercials or channel events might be recorded. To avoid annoying clips being recorded, an efficient and effective way to automatically locate the program segments is needed in a recorded television stream, for examplewe could build an enhanced TV program guiding the precise durations and locations of the programs. Such a program will be used by an end-user vcr, institutions that collect materialsfrom television and as well as channel themselves who wants to monitor what has been broadcasted. Another example is,companiesmight contract TV stations to broadcast their commercial advertisement.Then the companies would like to know whether the commercials are broadcasted following the terms in theircontracts e.g. when -before/after/during certain popular programs, and the exact durations and times, etc. Similarly, some other companies may contract survey companies to seek information about how their competitors market their commercials by monitoring competitors' broadcased TV commercials. Whilst the same commercial broadcasted in different TV stations, it will actually appear differently with some variations, for example station's own broadcasting parameters settings (e.g., frame rates, aspect ratio and resolution), and some insertions(e.g., station/company logos or contact information). As a consequence, various versions of the commercial video clips, which are broadcasted on different TV stations at different time, are likely to coversimilar content.Accordingly, commercial monitoring is needed and content-based near duplicate video detection couldbe a solution, though it still remains a challenge.

Content-based near-duplicate video detection has been studied for a decade in solving problems such asTV broadcast monitoring, video copyright enforcement, video database purge, cross-modal divergence detection, and have achieved some good results, though it still remains a challenge. In the literature, there are mainly four steps for near-duplicate video detection: i) video segmentation

(keyframeextraction); ii) video content summarization and representation (feature extraction); iii) similarity measure/matching; iv) near-duplicate video recognition.

### *Web Video Control and Management*

Broadband internet access is increasingly common with the rapid advancement of technology. Especially, with the popularity of social media in Web 2.0, videos that carry the richest content for daily information communication are available on the internet. They are spreading on the internet with exponential growth that brings challenges for web video database control and management. With the advances of hardware, the cost of storage is dropping, and users can obtain web videos easily toredistribute them again with some changes by using popular video edit software which can be downloaded online for free. Users all over the world are uploading edited/non-edited videos and searching for videos on a daily basis. Consequently, online video collections are rapidly becoming larger, in no small part due to numerous duplicate or near duplicate videos being collected. With millions or even billions of videos, collections are very difficult to control, manage, maintain and search. Therefore, investigation of effective content management approaches plays an important role in improving the management of large scale video collections and also enhancing the performance of content-based low level video retrieval. Automatic video-content-based clustering is one way to address this topical issue, facilitating organization to improve the effectiveness and efficiency of video document access.

In addition, broadband access tomedia isfaster and cheaper, videos are becoming very popular on the web. For example, 65,000 new videos are upload to the video sharing website YouTube everydayand the daily video views are more than 100 millions [1]. According to comScore [7], a leaderin measuring the digital world, reported that U.S.internet audiences viewed more than 9 billion online web videos in July 2007 alone. The average online video viewer consumed 68 videos

during the month which is more than 2 videos per day, and the figure is still expected to keep climbing. In July 2008, the U.S. internet audience viewed more than 11.4 billion online web videos alone, and the average online web video viewer consumed 80 videos during the month[8]. YouTube dominates online video providing in the United States, with a market share of around 43 percent and more than 14 billion videos viewed in May 2010 [9]. YouTube says that 35 hours of new videos are uploaded to the site every minute, and around 75 percent of the material comes from outside the U.S. Mostly, videos in youtube are posted without sufficient tags and managed effectively. Moreover, some are even posted without any tags or descriptions. Clearly, this shows us the fact thatonline videos are substantially growing rapidly with very similar content [23].The recent online web video view ranking released by comScore said, 180 million US internet users watched 36.6 billion online content videos in May 2012 [114]. In April 2013, 181.9 million Americans watched 38.8 billion online content videos [115].From the increasing figures in different years, it is clear to see that the rapid advancein the technology generation andgrowth of propagation of digital videos in both centralized video archives and distributed video resources on the Web has brought us an urgent need for video search engines to efficiently retrieve relevant videos of interest.

With the huge number of video uploaded through the internet, the duplicate and near duplicate videos are collected in the video database remain undetected by current text annotation technology.Text annotation is the traditional solution for search engines. This solution is inefficient and ineffective: 1) text tagging relies in human’s subjective perspective view. However, different people have different perspective views on video semantic meaning, which will cause the duplicate or near-duplicate videos to be tagged differently from different people and places

around the world and uploaded again. Due to human's limited semantic description, those duplicate/near-duplicate videos cannot be detected and some maybe overwhelmed when users search videos by typing semantic meaning of video which also has a different perspective and limited text description. 2) The dramatic increasing number of videos makes manual text tagging very tedious. With the huge amount of videos uploaded every day, we can imagine how much manpower we need for the job of manual text tagging of the huge amount of video collection, which is rather time consuming and tedious. 3) Hard to automatically categorize the videos with tagged text, since tagged text does not fully express the semantic meaning of video. Therefore an automatic and efficient way to manage videos is highly demanded. To avoid getting swamped by almost identical copies of the same video while searching, efficient NDV detection and elimination is significant for effective and efficient search, retrieval. Video classification or annotation is an open research problem. However, it still remains challenge and not robust enough for applications.

The uploaded duplicate or near-duplicate videos wastes a large amount of storage space, increases the storage expense and makes database management and control more complicated and difficult.Submitting a text query, the web video search engines tend to return a list of ranked search results according to their relevance scores. Sometimes retrieved relevant information items that meet user's requirements are ranked topmost, while sometimes few relevant information is retrieved, which does not meet to users' requirement. Moreover the topmost search results mostly contain a great deal of redundant similar videos.Based on a sample of 24 popular queries from YouTube [3], Google Video [5] and Yahoo! Video [4], on average there are 27% redundant videos that are duplicates or nearly duplicates to the most popular version of a video in the search results [44]. This statistics of

redundant videos information was stated in 2007. With the rapid video uploading and consuming as the figures in year 2007, 2008 and 2010 shown above, we can imagine that the online redundant video information will increase.Banking on this observation, the need to develop tools to identify near-duplicate videos becomes more important.

Therefore, an important problem now faced by these video sharing sites is how to automatically perform accurate and fast similarity search for an incoming video clip against its huge database, to avoid copyright violation. Meanwhile, the retrieval efficiency will be hampered if there are a large number of search results returned almost identicalcontent; accordingly, database purges also contributes to high-quality ranking to improve the video retrieval results [44]. The traditional way to protect copyright by using embedded watermark into videos. However, not every video has a copyright watermark embedded, this brings us to the difficulty to monitor whether the videos are distributed by others illegally e.g. where and how they redistribute the videos. Also, a watermark is inefficient due to manually embedding the copyright information which is tedious work.

## 1.2. Motivations

NDV retrieval has been explored for a number of years. The goal of this study is to maximize the retrieval accuracy in minimum retrieval time. Traditionally, researchers attempted to approach the goal through 3 aspects: i) the video representation; ii) matching scheme iii) indexing structure. These 3 aspects have been thoroughly and extensively studied in the literature.

Therefore, it is significant to seek for the improvement in other aspects. Since it is retrieving videos that are near duplicate, why not organize and group near duplicates together in the same cluster offline first before online retrieval? In such a

way, it will be easier to access and directly reduce the online retrieval searching space through offline pre-processing. Instead of accessing all videos, it only has to access the cluster representatives. Theoretically, itcan see that the access time is reduced by such an offline process. Meanwhile, with the access time being reduced, if the retrieval accuracy is enhanced or at least maintained, then pre-process dataset offline will be a great assistance in the enhancement of NDV retrieval performance. Moreover, improving NDVR in such a way has not been studied and explored yet.

## 1.3. Research Questions &Objectives

This section will pose some research questions and research objectives. The unsupervised clustering techniques have not been explored in order to improve retrieval performance in terms of accuracy and speed. The main task is to explore whether clustering techniques is feasible to assist in improving NDV retrieval performance.

### 1.3.1. Research Questions

In the literature, the indexing scheme is the main factor to reduce the searching space for achieving the fast retrieval speed, which is fulfilled on-fly. However,is there such a way that could reduce the searching space through offline process to further increase the retrieval speed? The purpose of NDV retrieval is to retrieve all NDV of the query video from the database, then before that, why not preprocess the database offline by using clustering algorithm to group NDVs in the same cluster to reduce the searching space and facilitate accessing and indexing. With this assumption, the following Research Questions (RQ) and Research Objectives (RO) have been identified.

*RQ1: Are clustering algorithms from the literature good for preprocessing NDVs?*

*RQ2: What is the impact of various clustering algorithms from the literature on NDV retrieval?*

*RQ3: Since video is always represented as a sequence keyframe's descriptor in the order of time, how to develop a clustering technique that caters to the video sequence descriptor?*

*RQ4: What is the impact of various video features (ordinal, global and local features) impact on MSA clustering-based NDV retrieval?*

*RQ5: What is the performance of MSA clustering-based NDV retrieval?*

### 1.3.2. Research Objectives

In this thesis, the objective is to study whether it is feasible and promising to improve NDV retrieval performance through offline dataset process, which could be achieved by utilizing unsupervised clustering techniques. The processed dataset will be clusters with NDVs grouped in the same cluster. The grouping is approximately depending on how the clustering algorithms perform.The ideal case is that all related NDVs are exactly grouped in the same cluster without any noise. However, there is no such clustering algorithm that can achieve that yet. After the clustering, the retrieval operation will be conducted on the processed dataset, which are clusters. The objective is to take advantage of the pre-processing dataset by using clustering algorithms to benefit the retrieval operation in terms of retrieval accuracy and speed.

*RO1:To study various clustering algorithms from the literature, and analyze with illustrations.*

*RO2: Experimental study the impact of various clustering algorithms from the literature on NDV retrieval compared to the naïve non-clustering based NDV*

*retrieval. Through the empirical results, it will show the knowledge of which pre-process clustering algorithm is better for NDV retrieval.*

*RO3: To propose a novel MSA-based clustering algorithmthat caters to video sequence signature representation, since the clustering algorithms are good for data points input only that are single vectors with the same length.*

*RO4:To evaluate the MSA clustering based NDV retrieval in terms of retrieval accuracy and speed compared to that of non-clustering based naïve NDV retrieval and also against other clustering methods in literature.*

*RO5: To experimental study how various video feature representations (ordinal, global and local features) impact on MSA-clustering based NDVs retrieval.*

## 1.4. Contributions

To achieve the objectives, this thesis has evaluated clustering-based NDV retrieval, and the empirical results show whether the clustering-based NDVs retrieval is feasible and promising. The contributions are listed as following:

- The impact of various clustering algorithms in literature on clustering-based NDV retrieval is experimentally studied.

  This thesis analyzes the clustering techniques in literature with illustrations. Then it processes video into 24-dimensional vector and apply K-means, DBSCAN, BIRCH, CURE, PROCLUS and EM clustering algorithm respectively to form clusters. The quality of clusters affects the retrieval performance. The experimental result shows how these clustering algorithms impact the retrieval performance.

- This thesis proposes a novel clustering algorithm based on Multiple Sequence Alignment (MSA), which could preserve video temporal information during the clustering operation to solve the problem that existing clustering algorithms which caters to the video sequence representation cannot be found in literature.

  Video is processed into a sequence of alphabetical DNA-like representation, and then apply MSA clustering method to group NDVs into clusters that, the cluster assembles similar videos.

- This thesisevaluates the effectiveness ofNDVR incorporating clusters formed by MSA clustering method. The evaluation shows the impact of various video feature representations on MSA clustering-based NDVR.

  In this evaluation, it considersordinal feature, global feature (the color, texture) and local feature (SIFT) for clustering-based NDVR. The experimental results show that both retrieval accuracy and speed are greatly enhanced by using color and texture for clustering-based NDVR compared to that of non-clustering based approach under the fair comparison criteria. The same as SIFT feature, the retrieval speed is dramatically increased while the retrieval accuracy maintains no dropping. By using the ordinal feature, besides comparing to non-clustering methods and dynamic programming and n-gram methods, it also compares to the NDV retrieval which incorporates various clustering algorithms to preprocess the dataset.

## 1.5. Thesis Organization

Chapter 1 introduces why content-based near duplicate video retrieval is important and what motivations, objectives, and contributions of this thesis are.Videos carry the

richest content information, and have become the most popular media to spread the information and share the information. The number of videos uploaded to websites is increasing exponentially, and a huge number of near-duplicates are collected. Thus, to investigate an efficient and effective way to retrieve near-duplicates is important.

Chapter 2 gives a review of how researchers improve retrieval accuracy and speed through video features, matching methods and indexing structures. Then several recent remarkable retrieval frameworks which are verified on a large-scale dataset are introduced. The retrieval accuracy is not the only requirement, but also retrieval speed. In recent decades, retrieval accuracy draws more attention than speed.

Chapter 3 gives the clustering-based NDV evaluation framework, and then goes through the clustering techniques by illustrating with real video features. The advantages and disadvantages of each clustering technique are analyzed as well. Finally, the experiment is studied of the impact of variousclustering techniques on the clustering-based NDV retrieval.

Chapter 4proposes the novel clustering framework based on multiple sequence alignment. The proposed framework caters to the video feature representation in the form of sequence with various lengthsin video time order.

Chapter 5 makes use of the formed clusters by using MSA clustering framework proposed in Chapter 4 for clustering-based NDV retrieval evaluation. Instead of comparing a query video to all candidate videos in database, the clustering-based framework compares to the representative or centroid of each cluster only. The different features may affect the clustering-based NDV retrieval as well. Therefore, an empirical study ofMSA clustering-based NDV retrieval by usingordinalfeature, global features (color and texture) and local feature (SIFT) is conductedrespectively and compared to that of non-clustering based naïve NDV

retrieval method. The results show that even by using different features, MSA clustering-based NDV retrieval outperforms that of non-clustering based under the same criteria.Moreover, by using ordinal feature, NDVR with MSA clustering method incorporated is also compared to that of various clustering algorithms incorporated for pre-processing the dataset.

Chapter 6 makes the conclusion of the overall thesis and future works.

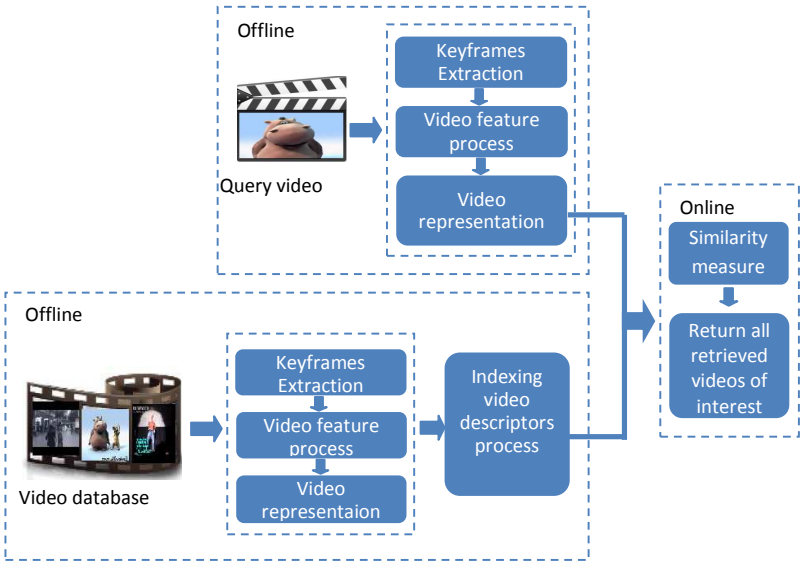
## 1.6. Summary

In this chapter, the research problemwas identified. Then it showed the motivations and objectives of this thesis, as well as the entire structure of this thesis.In the next chapter, the related works and recent critical contributions to the near-duplicate video retrieval in large-scale database will be studied.

# CHAPTER 2

# *RELATED WORK AND RECENT PROGRESS*

Traditionally, NDV retrieval is improved using mainly 3 aspects as shown in fig 2.1: i) Video representation. ii) similarity/distance measurement. iii) Indexing structure. Section 2.1 gives a literature review of NDV retrieval in these 3 aspects respectively.




**Figue 2.1. Traditional NDVR framework**

# 2.1. Related Work

## 2.1.1. Video Features

As we know, video consists of a sequence of frames (images) in time series order. Hence, a variety of image features have been studied to describe the video representation and compute the similarity between videos, e.g. color histograms [41][18][52], motion [35][52], intensity histograms [18], texture, and shape. However, the high-dimensional nature of the representation is always the concern of computing complexity. Kobla et al.[24] addressed the issue by exploring to represent the video descriptor in low-dimensions. Many other works have been contributed to the field study.

In this section, a brief review of features that have been studied in literature is given. Although a variety of features have been investigated to represent a video, there is still no feature in literaturethat can represent and describe the information of a video sufficiently and thoroughly without any information loss. To solve this problem, some researchers overcome the drawbacks of using a single feature by fusion of different features. However, different categorizations of videos require thefusion of different features, which brings another problem that, which fusion is the best to fit the categorization of videos?The current techniques are still very limited to address all the problems such as compact, robustness, low-dimensions etc to represent the video. Therefore, to have a compact feature sufficiently representing the video still remains a challenge.

### *Color*

This section will present how color was applied for NDV retrieval and detection. Color is one of the most popular and important features used to analyze the frame, though it is not fairly discriminated. Lienhart et al.[25] uses the color coherence

vector as the video fingerprint, which is a color histogram. However, besides computing the number of pixels of the certain color, it also differentiates pixels of the same color depending on whether they belong to the same region as coherent and otherwise as incoherent. In [48],Cheung &Zakhorpartitions the individual frame into four quadrants which incorporates spatial information, and then represents each individual frame by using four 178-bins color histograms in the HSV color space resulting in 712 dimensions in total to describe the single frame. Hoad&Zobel[49] uses a sequence of values to represent the video signature, which is based on frame color-shift. Either Manhattan distance or Euclidean distance could be applied to compute the color distance between two adjacent frames and form a sequence of numerical distance values to represent the video.Li et al. [50] present a binary sequence representation based method by incorporating the techniques of ordinal and color measurements. Regarding the color representation, each individual frame is quantized into a fixed number of colors, and each color element is represented by a 3-bins binary string which corresponding to the percentage range of pixels dominant the color element. Wu et al. [44]filter out the near-duplicate video using a hierarchical approach. In the initial coarse filtering, the global color histogram feature is employed to filter noise in a fast process. Although the color histogram is easy and fast to be derived, it lacks discrimination. Therefore only when a video cannot be clearly identified as novel or near-duplicate by using global signatures, the authors apply a more expensive local feature for near-duplicate video retrievalin the fine measurement phase. Since a local feature is rather expensive to compute and difficult to be scaled for NDV retrieval in real-time in large-scale databases, the introdution of global feature for coarse filtering directly reduces the searching space and uses local features to search candidates in a small dataset. The global color histogram lacks addressing the spatial character within frame, and is fragile to frame transformation, color distortion etc. Color

feature such as RGB, HSV, color histogram etc. is an easy way to describe frame information. It is infeasible to detect when the content has only slight color difference as well as too much noise will increase the detection difficulty and affect the detection accuracy.Sometimes there are many different objects or scenesbut described by the same color, in such a case, color becomes indiscriminate.

### *Ordinal*

This section will show ordinal feature used in the literature for NDV retrieval and detection. The ordinal features construct the frames spatial relationships. The frame is usually partitioned into NxN blocks and then will be ranked based on the average feature value of each block.Thus, it is insensitive to the ranking distortion.Bhat&Nayar[51]are the first who proposed ordinal signature for stereo correspondence. Hampapur et al. [52] compared the performance of video sequence matching by using ordinal feature, motion feature and color feature respectively. They concluded that ordinal feature performed better than two other features. Li et al. [50] presented a binary signature based method by combining the techniques of ordinal and color measurements.Regarding ordinal features, the authors partition the frame into NxN windows. The average intensity of the window is computed and then ranked. In [53], Hua et al. proposed a novel global visual feature which combines the spatial-temporal and the color range information representing in 144 dimensions space.Hua et al. partition the frame into 3x3 blocks and the rank is assigned to each block according the average gray level. Yuan et al. [32] propose a fast video descriptor representation based on ordinal measure of re-sampled video frames, which is robust to compression formats changes, compression ratios, frame sizes and rates. Ordinal measures have been proven to be robust to change in brightness, compression formats and compression ratios[54]. Compared to local interest point descriptor, ordinal signature outperforms in terms of complexity and

produce promising resultsthat are robust to some frame character changes. Ordinal features operate on NxN equal partitioned blocks of the frame. The feature of each block is extracted and ranked for further matching. This method is evaluated having a better accuracy performance than that of global features such as color. Although local interest points still outperforms ordinal feature regarding to variations in frame such as scaling, editing, crop etc, ordinal feature outperforms local interest points in terms of complexity and storage. In addition, its detection accuracy improved compared to global color feature.

### *Video Temporal Information*

This section will introduce how video temporal information is preserved for NDV retrieval and detection. It is the unique information of video that the image doesn't have. Being aware of preserving spatial character of image is not sufficient for video, many papers represent videos by preserving the temporal information of video. Adjeroh et al. [57][58], Chang & Lee [56],and Yazdani&Ozsoyoglu[55] have directly coded the temporal information by treating the video data in its natural form of a sequence of ordered frames. Dementhonet. al. [59] describes how to use video strands to create spatio-temporal descriptions to represent a video stream. Muneesawang& Guan[60] make use of temporal information to propose the template-frequency model. In [61], Kim et al. fuse the ordinal and temporal information together for sequence matching to enhance the video retrieval speed. Many other papers have made use of temporal information to represent video seeking for retrieval accuracy enhancement. Temporal feature characterizes the nature of video frames in an event happening order and conserves the temporal information, which makes videos more discriminate. Thus temporal information plays an important role to describe a video.

### *Fusion of Different Features*

The work of fusion of multiple features will be shown in this section. Analyzing the video with a single feature has been seen its drawbacks of description with insufficient information. Therefore the combination with multiple features is studied. In [53][59], the authors combined both spatial and temporal information to describe the video. Wang & Ngo [62] explore a variety of visual and audio analysis techniques in selecting the most representative video clips for rushes summarization at TRECVID 2007. The random combination of multiple features might not bring better results than a single feature. Different categorization of videos has different characteristics. Therefore, to investigate and analyze different group of videos is important to the selection of multiple features to represent a single video.

### *Local Interest Points Descriptor*

This section will introduce the work on local interest points descriptor. Local interest point descriptor is the most robust feature, which is invariant to various transformation, scale, or editionetc. of the image. However, each image could have thousands of interest points with high dimension detected, which is costly for comparison and storage of large amount of points for each frame in a video. Therefore, some index strategies are employed for indexing points to accelerate the retrieval, while some others combine the cheaper global feature together with more expensive local interest point descriptor. Joly et al. [63] detected the local interest point and compute the20-dimension descriptor from it. In [64], the authors made a conclusion that local descriptor performs better than ordinal measure to discriminate videos when there are captions inserted. Chum et al. [65]proposes and compares two novel schemes for near duplicate image and video-shot detection and retrieval. The first scheme applies global hierarchical color histograms, and theLocality Sensitive Hashing (LSH) is used to index features for fast retrieval. The

second scheme uses local feature descriptors (SIFT) for better video identification and the min-Hash algorithm is exploitedfor retrieval techniques which has been applied to compute approximate set intersections between documents in the information retrieval community. Local interest points descriptor such as SIFT, SURF etc. are the most robust feature. The drawbacks of this feature is the difficulty to index and store the large set of interest points detected in one frame which are up to the amount of thousands. To compare considerable number of interest points of single frame description, it requires exhaustive computational complexity. The most recent work is to use Bag-of-Word (BoW) method [44], which construct the visual words code book and then map each local point to the visual words. A video is then represented by a bag of visual words. The high dimensional points are then converted to the texts, which is easier and faster for retrieval.

### *Other Features*

Some video feature summarization techniques are extensively investigated. Due to robustness and complexity limitations of above mentioned signatures, researchers have tried many approaches to summarize the feature based on model or pattern in order to obtain a compact signature that is efficient and effective for locating NDVs in large-scale database.Cheung &Zakhor[48]sampled a small set of frames and then summarizes each video according to the sample;the method is called the video signature (ViSig). Again in 2005, Cheung &Zakhor[66] presents how to use the ViSig method torepresent the video as the high-dimensional feature vectors and propose a novel nonlinear feature extraction technique on arbitrary metric spaces that incorporate the triangle inequality and the classical Principal Component Analysis (PCA) techniques. There are also other video summarization techniques for similarity measurement such as density parameterization[67] based techniqueor

producing keyframesby using k-means or k-medoids algorithm [68].Compared to these techniques, ViSigperforms better in terms of computational complexity thatthe only a single pass of a video sequence to compute its signature requires much lower computational complexity.Zhou & Zhang [69] represented video as normalized chromaticity histogram which is illumination-invariant. In [70], the authors compute image signatures from the low frequency coefficients of the Discrete Cosine Transformwhich is an integer value. The authors in [71] introduced a graph-based method, which detects objects in each frame of thevideo, and the moving objects of each frame are then indexed by STRG-index method in the time series order. Shen et al. [73] outlines a system for detecting near-duplicate videos by using a novel statistical method to summarize the content features for each clip. In [74], a new system for video summarization called “Near-Lossless Video Summarization” is developed. The system is able to summarize a video stream with the least information loss by using an extremely small piece of metadata. The summary consists of a set of synthesized mosaics and representative keyframes, a compressed audio stream, as well as the metadata about video structure and motion. Although at a very low compression ratio (i.e., 1/30 of H.264 baseline in average, where traditional compression techniques like H.264 fail to preserve the fidelity), the summary still can be used to reconstruct the original video (with the same duration) nearly without semantic information loss.

### 2.1.2. Video Matching

This section will introduce the work on video similarity/distance measurement methods. The frame-based approach is the simplest and most exhaustive method for video similarity measure. The method compares every individual frame between two video clips in order to identify thesubsequence of frames that are consistent orsimilar[13][25][43][40][36]. Cheung &Zakhor [48]introduced an

efficient similarity detection algorithm for their video signature. The video clip signature is computed by selecting a small set of its frames that are most similar to a set of random seed images samples. The statistical pruning algorithm is then introduced for fast near duplicatevideo detection/retrieval in very large databases. Euclidean distance and sequence alignment are the most common and widely appliedfor measuring distance between image vectors and video sequences respectively. However, they do not cater to any feature. Therefore many other measurement methods are proposed to fit their own feature distance measure.

### *Hausdorff Distance*

This section will present the work on Hausdorff Distance for distance measurement between videos. In [68], the authors proposed to measure the maximal dissimilarity between shots. Kim & Park [75] proposed the novel matching algorithm which reduces the computational complexity by usingthe modified Hausdorff distance. Moreover, the novel approach of computation of the similarity between keyframeswas proposed to improve the performance in terms of the accuracy. The hausdorff distance is usually applied on the sets. This distance measure method is not broadly applied on video analysis.

### *Sequence Alignment*

This section will show the work on sequence alignment for distance measurement between videos. Sequence alignment methods have been widely utilized to measure the distance between two videos. A video is a sequence of images, it can be easily represented by a sequence of string/value or a sequence of symbols, which consist of a sequence of consecutive image representations. To measure distance of such sequences (two videos), sequence alignment methods such as dynamic programming, edit distance, local alignment etc areintensively studied. In [57],

local alignment was used to identify relevant regions within large video clips. In [49], Matching is processed by applying local alignment forlocating sequences with similar values between video clips. Tan et al. [76] proposes a distance measure to evaluate the video similarity by aligning the frames of two video sequences. In [69], authors applied Dynamic Programming (DP)on video shots to find the optimal nonlinear mapping between video sequences. Edit distance for video similarity/distance measure has been widely studied, and was initially proposed in [79]. In [78], Bertini et al.measure similarity between videos by using an edit distance. The purpose of edit distance is applied to obtain high similarity between videos that have difference in some subsequences, but are essentially related to the identical visual content. Guimaraes et al. [80] propose Boyer-Moore-Horspool, which is a modified fastest exact string matching algorithm to compute video clip repetitions for similarity measure. In [79] [81], they used the longest common substring algorithm by using dynamic programming. Zhou et al. [82] proposes a new similarity measure, named as Video Edit Distance, which adopts a complementary information compensation scheme based on the visual features and sequence context of videos. Though dynamic programming is a costly algorithm, it has a good performance to measure distance between two video sequences. Sequence alignment distance measure method is extensively studied for video distance measure due to the nature of video that video can be easily represented as a sequence of string. Dynamic programming shows a good performance in terms of accuracy. However, it is fairly expensive to compute.

### *Some Other Novel Matching Techniques*

There are many well-known distance measure methodsnot satisfied by the high dimension video features due to thecomputational complexity and poor scalability. Many novel distance measures are proposed to identify relevant videos in large

scale database as well as tackle the problem of video comparing in high-dimensional space. In [43], the authors measure Voronoi video similarity in large-scale video database by giving the intersection volume between voronoi cells of similar clusters.Similarly, Video Triplet is another video similarity extraction framework, which models the similarity as a composition of position, radius, and density for a cluster with a tightly bounded hyper sphere [72]. In [83], stage starts from acoustic matches and validates the hypothesized matches using the visual channel. Finally, the precise segmentation uses ne-grain acoustic match poles to determine start and end-points.

### 2.1.3. IndexingStructure

This section will present the work on indexing structure for the purpose of achieving fast retrieval speed. Numerous smart indexing structures and searching schemes have been proposed in order to efficiently locate targets in the high-dimensional spaces [37] [32].Kashino et al. [45] proposes a fastsearching method to search similarity-based signals for fast locating the query that is represented in the form of a specific audio and visual signal.The pruning algorithm plays an important role in accelerating the retrieval. Through the pruning, the searching time could be reduced up to hundreds of times compared to the exhaustive searching. To detect or retrieve near duplicate videos in a large-scale multimedia database, the indexing structure is a major factor to guide the searching and reduce the searching space for assisting in locating the target in a fast fashion.Some well-performing approaches such as Locality-Sensitive Hashing (LSH)are widely applied for indexing vectors. Also inverted File is intensively explored to index SIFT bag of words features.

*File Structure*

This section will introduce the work on file structure for indexing. File structure has been extensively studied for video corpus indexing. For example Lu et al. [84]address the problem of content-based video indexing by proposing the Ordered VA-File (OVA-File) based on the VA-file. Shang et al. [85] leverages relative gray-level intensity distribution in a frame and temporal information of videos along frame sequence. The new index structure is proposed utilizing an inverted file for fast histogram intersection computation between videos. The file indexing stores the mapping from content to its locations or file addresses in database. The structure is usually applied for document retrieval, and it is easy to index large-scale databases.

### *Hashing*

This section will introduce hashing indexing structures. Hashing is another commonly used scheme for indexing. For example Oostveen et al. [86]used a lookup table to store features which are based on the average of luminance of the frame blocks. Pua et al. [87]proposed a hashing structure to index color moment vectors for efficient retrieval of repeated video clips. This isthe same as Naturel &Gros[70] who use hash table to store the feature representations.Ke et al. [88] andYang et al. [89]employ LSH to index the local descriptors. Zhao et al. [90]use LSH to index color feature and visual keywords. LSH is the most efficient indexing structure that is often applied to index large database with vectors. In hashing indexing structure, the hashing function is the key to determine the indexing accuracy. Videos that have the same hashing value will be sent to the same hashing bin. The accuracy of the hash function determines the effectiveness and efficiency of the hashing indexing.

### *Tree*

Tree indexing scheme will be shown in this section. Various trees have been widely applied for indexing high dimensional features in order to reduce the searching space. For example Lu et al.[91] review some early works on clustering methods and multidimensional indexing structures, such as k-dimensional tree, R-tree, and Telescopic-Vector-tree, etc. Park & Chu [92] propose suffix tree indexing techniques for video and image sequence matching process. Zhou et al. [82] propose a compact video representation summarized by usingthe global feature. Each video in a database is mapped into a digital string(a series of cluster id). Optimal B+-tree is employedto identify similar clusters and an inverted list is attached for quickly locating potentially similar videos. In [93], the frames in a video are ranked according to their similarity on the distribution of salient points and color values. Then, a tree based approach is used to seek for the repetitions of a video sequence. There are many different hierarchical trees available for indexing purposes. The tree guides searching in one of the all possible paths by selecting one of tree branches. However, the implementation of trees is complicated; especially when there are more than 2 branches for each tree node. Besides, it is rather memory consuming to construct a large tree.

### *Filtering Scheme*

This section will present the work on filtering scheme. Besides those smart indexing algorithm and scheme, a two step coarse-to-fine filtering strategy is another option to accelerate locating similar target videos. The less expensive global feature is usually utilized for roughly filtering out a large portion of unrelated videos information in the coarse phase, and in thefine step, the more expensive or more robust local features are employed for the comparison. E.g. Hoi et al. [94] propose a two-step filter-and-refine approach based on nearest feature trajectories. Hua et al. [53] developed a coarse-to-fine signature comparison

scheme. In the coarse searching step, roughly matched positions are determined based on Sequence Shape Similarity, while in the fine searching step, dynamic programming is applied to handle similarity matching in the cases of losing frames and temporal editing processes are employed on the target video. Zhu et al. [95] suggest an effective multi-level ranking scheme that filters out the irrelevant results in a coarse-to-fine manner. The two step coarse-to-fine filtering scheme is efficient, because the coarse filtering roughly filters out most of theirrelevant information. The less expensive global feature is applied in this stage for speeding up filtering. In such a way, the searching space will be dramatically reduced in the first filtering. When there are fewer candidates left, the fine filtering helps locate the target in an accurate manner.

### *Novel Indexing Scheme*

Meanwhile, researchers have proposed other novel indexing schemes, for exampleValle et al. [96] introduce multicurves, a new scheme for indexing high dimensional descriptors. This technique, based on the simultaneous use of moderate-dimensional space-filling curves, has as its main advantages the ability to handle high-dimensional data (100 dimensions and over), to allow the easy maintenance of the indexes (inclusion and deletion of data), and to adapt well to secondary storage, thus providing scalability to huge databases (millions, or even thousands of millions of descriptors). To improve the effectiveness of the indexing structure, novel indexing structures such as multicuves are introduced to adapt the different feature descriptors for better locating target candidate videos in the searching space. Compared to traditional indexing schemes, novel indexing is less flexible in that is only adapted to specific descriptor structure or data.

## 2.2. Recent Progress

NDV retrieval has been investigated for decades. Video features, distance measure between two videos and indexing structures have been intensively studied in order to enhance the retrieval accuracy and speed. However, it still remains a challenge. This section, introduces several real-time retrieval models in recent years.

Huang et al. [97] proposed a Bounded Coordinate System (BCS) to search videos clips in real-time.Given a video clip $X = \{x_1, x_2, \ldots, x_n\}$, where $n$ is the length of the video. The Bounded Coordinate System $BCS(X) = (O, \ddot{\Phi}_1, \ddot{\Phi}_2, \ldots, \ddot{\Phi}_d)$, where d is the number of dimensions and $n \geq d$, $O$ (the origin of the coordinate system) describes the mean of $x_i$, and $\ddot{\Phi}$ denotes the segment of the line bounded by two furthest projected points by using PCA principle. Then the rotation, translation and scaling operations are applied to measure the similarity between two *BCS* coordinate systems. To scale up, the authors extend the $B^+$-tree to index two distance values andname it*Bidistance Transformation* (BDT). The*BCS* represent the video in a simple and compact way. However, it will lose most of video information by projecting points to the new coordinate system.

Shang et al. [98] proposedCE-based spatiotemporal feature and LBP-based spatiotemporal feature for real-time large-scale near-duplicate web video retrieval. Both methods are using binary, which preserves the spatiotemporal information, in order to achieve the real-time video retrieval. Inspired by the traditional ordinal method that ranks the intensity of frame partition blocks,CE-based spatiotemporal featureconcludes 36 relations between 9 blocks. The entropy formula is then applied to select and sample the subset of features that carries most information. The LBP-based spatiotemporal feature also uses relations between blocks. However, the relation is separated into central regions and marginal regions. According to the central and marginal mapping function, a frame is represented by

an eight bits binarystring which describes the central and marginal region relations. Then the inverted file indexing structure is applied to speed up filtering candidate videos, and the intersection kernelis introduced to measure the similarity between two videos. Since the proposed features are using binary bits, it costs less memory for indexing and computing similarities and also less storage space.

Song et al. [99]proposed Multiple Features Hashing (MFH) for real-time large-scale near-duplicate video retrieval. In this paper, the authors trained two features, global and local feature.The proposed MFH method comprises two phases. The first phase is processed offline, which proposes the MFH algorithm to learn a series of *s* hash functions to hash the video keyframe feature into *s*-sized hash codes. In the second phase which is an online processing, the query video is processed in the same manner by mapping keyframe features to *s*-sized hash codes via *s* hash functions. The efficient XOR operation on the hash codes could then be efficiently performed to measure the similarity between videos.

Cai et al. [122] applies K-Mean algorithms to cluster the sample keyframes of videos by using colorcorrelogram feature, which constructs the spatial relationship between colors. The keyframes of the Video are then mapped to the closest cluster to describe the video. As a result, the video is represented by BoW. Aninverted File is introduced to index BoW. The authors reported that less than a second is consumed to locate a single query video in a million videos database that is collected by themselves and the biggest database reportedso far.

## 2.3. Summary

This chapter surveyed the solutions proposed to solve the content-based NDV retrieval problem. Several well-known models proposed to retrieve NDVs in large-scale database are also introduced. Through the study, it is known that traditionally,

improvement of NDV retrieval through Video features, matching, indexing have been extensively studied. Thus, it is a challenge to seek for enhancement on these three aspects in a deeper manner.In the next chapter, to cater to the nature of variable video sequence feature representation, a novel clustering framework ofnear duplicate videos will be proposed to seek for the feasibility of clustering-based NDV retrieval compared to that of non-clustering based methods under the same fair criteria e.g. using the same feature, same matching methods, and same threshold.

# CHAPTER 3

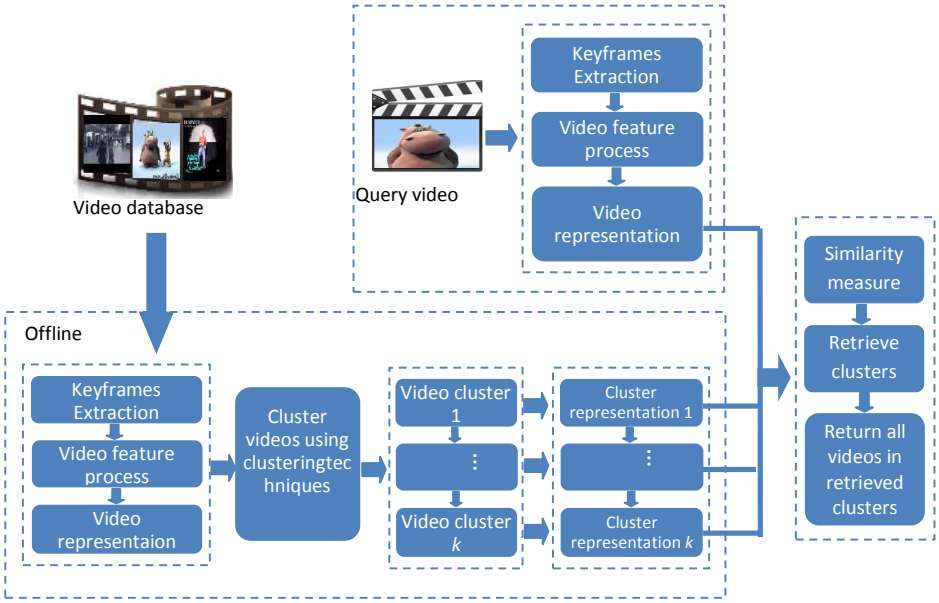
# *STUDYING THE IMPACT OF CLUSTERING TECHNIQUES IN LITERATURE ON NEAR-DUPLICATE VIDEO RETRIEVAL*

This chapter will study the impact of clustering techniques in literature on NDV retrieval. With clustering, the retrieval speed will be reduced without any doubt (c.f. Section 3.1). The experiment will show the clustering-based retrieval accuracy with various clustering algorithm incorporated compared to that of non-clustering based naïve retrieval method. Before the experiment, the evaluation framework will be presented in Section 3.1, and the clustering techniques in literature will be analyzed with illustrations in Section 3.2.

## 3.1. Evaluation Framework

Fig 3.1 presents the evaluation framework. First, the video dataset is processedfor video feature extraction. Second, it will apply the clustering techniques to pre-classify the dataset into clusters, and compute the cluster centroid or representative

video to represent each cluster. Third, it will evaluate NDVR with the clustered dataset and compared to that of non-clustering based approach. Instead of comparing a query video to all video candidates in the dataset sequentially (NDVR without clusters), it only has to compare to the representation (centroid) of each cluster (NDVR with clusters). Hence, it is necessary to conduct the fully experimental study to obtain the prior-knowledge of how the different clustering techniques impact the retrieval performance.




**Figue 3.1. Clustering-based NDVR evaluation framework**

It is believed that the proposed clustering-based retrieval is feasible and effective, since the clustering methods have the same effect as the retrieval operations; both are using some similarity measurement to decide whether they are NDV copies. The difference is that the clustering is operated offline, while NDV retrieval operation is online. Thus, this chapterwill consider incorporating various

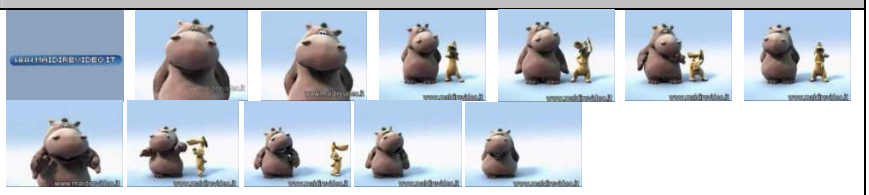
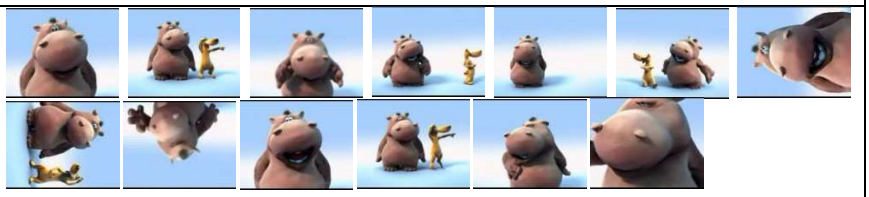
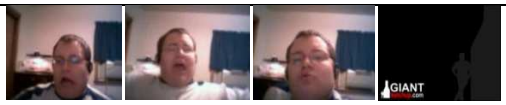
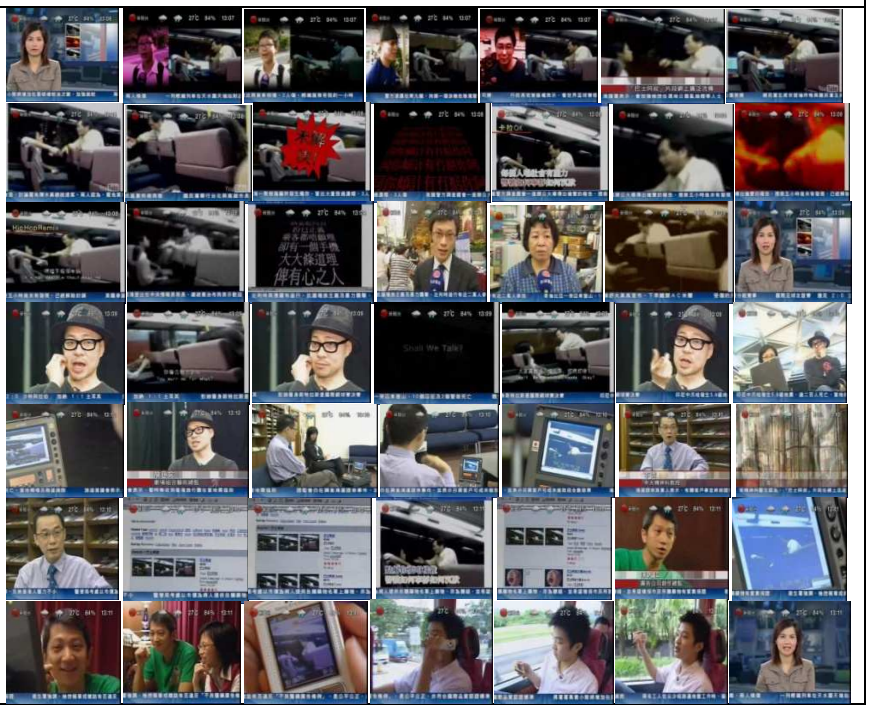
clustering algorithms in literature to pre-process dataset offline and evaluate what their impacts on NDV retrieval compared to that of non-clustering based naïve retrieval framework.

## 3.2. Clustering techniques presentation and analysis with illustrations

This section will brief the concept of six well-known clustering algorithms. There are six categorizations of clustering algorithms:

- Center-based clustering: construct various partitions and then evaluate them by some criterion. E.g. K-means clustering algorithm etc.
- Hierarchical clustering: create a hierarchical decomposition of the set of data using some criterion. There is an agglomerative approach and divisive approach. E.g. birch, cure clustering algorithm etc.
- Density-based clustering: based on connectivity and density functions. E.g. dbscan clustering algorithm etc.
- Grid based clustering: based on a multiple-level granularity structure. E.g. clique clustering algorithm etc.
- Model-based clustering: a model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other. E.g. EM clustering algorithm etc.
- Subspace-based clustering: this method is suitable for high dimensional data, which projects high dimensions into lower dimensions for clustering. E.g. Proclus clustering algorithm etc.

Before analyzing the clustering algorithms, the example videos (table 3.1) will be used in illustration and how their feature areprocessed will be presented (c.f. fig 3.2).

| Videos | Video clip keyframes |
|---|---|
| 1_45 |  |
| 1_3 |  |
| 22_11041 |  |
| 22_11042 |  |
| 18_9324 |  |

| | |
|---|---|
| |  |

**Table 3.1 . Video signature representation**

The videos are then processed for feature extraction (c.f. fig 3.2). Each keyframe of the video will be partitioned into 2x2 blocks. The average intensity of each block is computed, and the blocks are ranked according to these average intensity value. With 4 blocks of the keyframe, there are 4!=24 possible rankings by permutations. The 24 possible rankings will be regarded as the 24 bins of the video histogram. Each bin is the percentage of keyframes in the video falling in the corresponding bin.

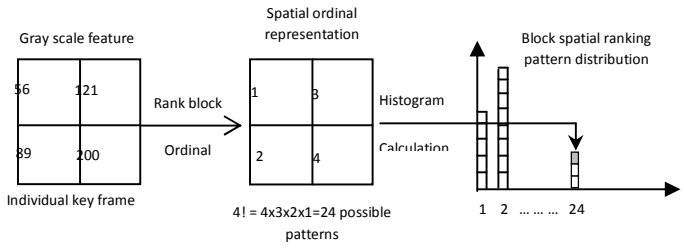



**Figure 3.2. Video spatial pattern histogram processing**

The extracted 24-dimensional video histogram features are shown in table 3.2.

| videos | Video signature |
|---|---|
| **1_45** | .0 .0 .0 .0 .0 .0 .0 .0 .1667 .0 .5 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0833 .25 |
| **1_3** | .0769 .0 .0 .0 .0 .0 .0 .0769 .1538 .0 .1538 .0 .0 .0769 .0 .0 .0 .0 .0 .0 .0 .0 .0 .461 5 |
| **22_11041** | .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .6667 .0 .0 .0 .0 .0 .3333 .0 |
| **22_11042** | .0 .25 .0 .0 .0 .0 .0 .0 .25 .0 .0 .0 .0 .0 .0 .0 .5 .0 .0 .0 .0 .0 .0 .0 |
| **18_9324** | .1633 .0408 .0612 .1224 .0816 .0 .0 .0612 .0 .0 .0408 .0408 .0204 .0816 .0 .0 .06 12 .0204 .0204 .102 .0 .0612 .0 .0204 |

**Table 3.2. Video signature representation**

### 3.2.1. K-MEANS

K-means [21] algorithm is one of the most classical and simplest unsupervised learning algorithms that solve the clustering problem. The algorithm introduces a simple way to classify a given dataset into a certain number of clusters, k clusters, which is given as the parameter. The algorithm follows the following steps:

[1] Choose k cluster centers to coincide with k randomly chosen patterns or k randomly defined points inside the hyper volume containing the pattern set.
[2] Assign each pattern to the closest cluster center.
[3] Recomputed the cluster centers using the current cluster memberships.
[4] If a convergence criterion is not met, go to step 2. Typical convergence criteria are: no (or minimal) reassignment of patterns to new cluster centers, or minimal decrease in squared error.

#### *Illustration*

In this section, it takes an example to go through the algorithm. Five real videos are selected from the dataset (c.f. table 3.1), and the video signatures are shown in table 3.2. There are 3 classes in this dataset. Group 1(video 1_45 and video 1_3), group2 (video 22_11041 and video 22_11042) and group 3 (video 18_9324).

***Iteration 1***

In the initial stage of the algorithm, k is set to 3, and then sample patterns are randomly selected. In this example, k sample patterns are simply selected sequentially which are video 1_45 as cluster 1, 1_3 as cluster 2 and 22_11041 as cluster 3. Then there are 3 clusters (table 3.3.) with the cluster centroid shown in table 3.4.

| | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|
| **Member videos** | 1_45 | 1_3 | 22_11041 |

**Table 3.3. Cluster result in iteration 1**

| | Centroid |
|---|---|
| **Cluster 1** | .0 .0 .0 .0 .0 .0 .0 .0 .1667 .0 .5 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0833 .25 |
| **Cluster 2** | .0769 .0 .0 .0 .0 .0 .0 .0769 .1538 .0 .1538 .0 .0 .0769 .0 .0 .0 .0 .0 .0 .0 .0 .4615 |
| **Cluster 3** | .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .6667 .0 .0 .0 .0 .0 .3333 .0 |

**Table 3.4. Cluster centroid in iteration 1**

The distance of all videos are computed to all new cluster centroids and obtain the results in table 3.5.

| | 1_45 | 1_3 | 22_11041 | 22_11042 | 18_9324 |
|---|---|---|---|---|---|
| **Cluster 1** | .0 | .4325 | .9204 | .7993 | .619 |
| **Cluster 2** | .4325 | .0 | .913 | .759 | .5354 |
| **Cluster 3** | .9204 | .913 | .0 | .5137 | .7489 |

**Table 3.5.Video distance to cluster centroids in iteration 1**

Secondly, assign the rest of the videos to the closest cluster based on Euclidean distance between the video and cluster centroid. According to the Euclidian distance table, video 22_11042 is closest to cluster 3, where the distance is .5137. Video 22_11042is assigned to cluster 3 then. In the same way, video 18_9324 is assigned to cluster 2. Then all the assignments are done, and cluster 1 (video members 1_45), cluster 2 (video members 1_3 and 18_9324), and cluster 3 (video members 22_11041 and 22_11042) are obtained as shown in table 3.6. After the assignments, the center of each cluster has to be recomputed.

| | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|
| **Member videos** | 1_45 | 1_3, 18_9324 | 22_11041, 22_11042 |

**Table 3.6. Reassign videos to clusters**

Continue to repeat the operation in iteration 1 until it reaches the specified number of iterations or it converges. In this case, there are two iterations only, and have the final converged cluster result in table 3.7.

| | Cluster 1 | Cluster 2 | Cluster 3 |
|---|---|---|---|
| **Member videos** | 1_45 | 1_3, 18_9324 | 22_11041, 22_11042 |

**Table 3.7. Final clustering result**

In iteration 2, the cluster result is the same as that of in iteration 1, which means the centroid and video memberships are not moving anymore. The iteration is terminated. The cluster result in iteration 2 is the final results.

### 3.2.2. BIRCH

Birch [19] consists of two key phases: 1) scans the database to build an in-memory tree. 2) Applies the clustering algorithm to cluster the leaf nodes. There are four steps required for inserting an entry into a clustering feature (CF) tree. CF information of the cluster is defined as a triple CF= ($N, \vec{LS}, SS$), where N is the number of data points (videos) in the cluster, $\vec{LS}$ is the linear sum of the N data points (video vectors), and SS is the square sum of the N data points (video vectors). Given entry “Ent”, it proceeds as below:

[1] Identifying the appropriate leaf: Starting from the root, according to a chosen distance metric D0 to D4 (c.f. appendix A for formulas), it recursively descends the CF tree by choosing the closest child node.

[2] Modifying the leaf: When it reaches a leaf node, it finds the closest leaf entry, and tests whether the node can absorb it without violating the threshold condition.

[3] Modifying the path to the leaf: After inserting "Ent" into a leaf, the CF information must be updated for each non-leaf entry on the path to the leaf.

[4] Merging Refinement: In the presence of skewed data input order, split can affect the clustering quality, and also reduce space utilization. A simple additional merging often helps ameliorate these problems: suppose the propagation of one split stops at some non-leaf node $N_j$, i.e., $N_j$ can accommodate the additional entry resulting from the split.

### *Illustration*

To keep the consistency and convenience of analysis, the same videos 1_45, 1_3, 22_11041, 22_11042 and 18_9324are chosen as example again (c.f. table 3.1, 3.2).

The non-leaf branching factor *B*=4. A leaf node contains at most L=3 entries. The diameter T=0.5. The branching factor *B*=3, video 1_45 is randomly picked to initialize the CF tree. Now it has the initial entry CF1 in root. The root node in this moment is the leaf node as well, since it is the only node in the tree at the moment. Then the rest of videos are inserted starting from the root, according to distance metric D2, and recursively descends the CF tree by choosing the closest child node (fig 3.3).

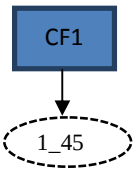



**Figure 3.3. Video 1_45 is inserted into the root**

It starts to build the root with entry CF1 (table 3.8.),CF1= (1, $\vec{LS_1}$, $SS_1$)

| | |
|---|---|
| $\vec{LS_1}$ | .0 .0 .0 .0 .0 .0 .0 .0 .1667 .0 .5 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0833 .25 |
| $SS_1$ | .0 .0 .0 .0 .0 .0 .0 .0 .0278 .0 .25 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0069 .0625 |

**Table 3.8.*LS* and *SS* information in CF1 tree node**

Now it shows how to insert video 1_3. Starting from the root, it measures the distance between 1_3 and all other CF entries in the root and chooses the closest entry. The D2 distance between 1_3 and CF1 is 0.4352. There are only 2 videos in the cluster, hence the diameter is equal to the D2 metrics of two videos, which doesn't violate the parameter condition T, therefore video 1_3 is absorbed by CF1 (fig. 3.4).

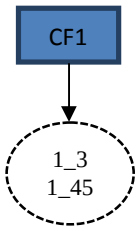



**Figure 3.4. Insert video to CF1**

After absorbing one more video, The CF1 information has to be updated by the linear addition of the two video vectors and the square sum.

After that, it has to update the CF information up this path (table 3.9),

CF1= (2, $\vec{LS_1}$, $SS_1$)

| | |
|---|---|
| $\vec{LS_1}$ | .0768 .0 .0 .0 .0 .0 .0 .0769 .3205 .0 .6538 .0 .0 .0769 .0 .0 .0 .0 .0 .0 .0 .0833 .7115 |
| $SS_1$ | .0059 .0 .0 .0 .0 .0 .0 .0059 .0515 .0 .2737 .0 .0 .0059 .0 .0 .0 .0 .0 .0 .0 .0069 .2755 |

**Table 3.9. Update CF1 information after insertion**

In the same way, it descends the video 22_11041 for insertion from the root to the leaf node. There is only one CF node, thus it does not have to search for the closest entry. It only has to check whether video 22_11041 could be absorbed by leaf node CF1 by comparing diameter=0.7562 to the threshold T=0.5, which violate the threshold condition and cannot be absorbed by CF1 entry. The number of entries in leaf node hasn't reached the L condition yet. Thus a new entry is created in leaf node called CF02 and video 22_11041 is absorbed by entry CF02 (fig 3.5).

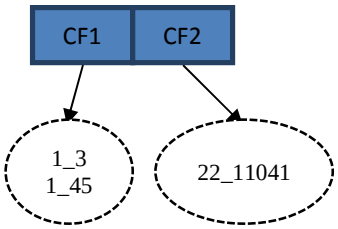



**Figure 3.5. Create CF2 and insert videos into tree**

And update the CF02 information (table 3.10), CF2= (1, $\vec{LS}_2$ , $SS_2$)

| | |
|---|---|
| $\vec{LS}_2$ | .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .6667 .0 .0 .0 .0 .0 .3333 .0 |
| $SS_2$ | .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .4445 .0 .0 .0 .0 .0 .1111 .0 |

**Table 3.10. Update CF2 information**

Video 22_11041 is inserted into the tree in the same fashion, and the result tree is shown in fig 3.6.

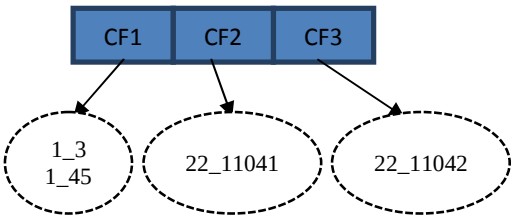



**Figure 3.6. Create CF3 and insert videos**

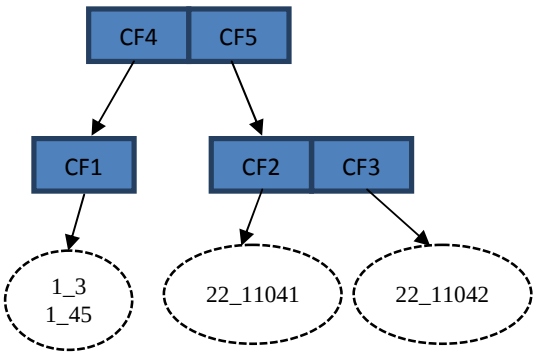



**Figure 3.7. Split tree result**

Finally, it inserts the last video 18_9324. It cannot be absorbed by any entry, since it is violating threshold T in all clusters. There is no space for creating a new entry in leaf node as well. The node has to be splitted by choosing the farthest pair of entries as seeds, and redistributing the remaining entries based on the closest criteria. After splitting, the initial CF tree is obtained as shown in fig 3.7.

The CF vector information is updated again as:CF4 = CF1= (2, $\vec{LS_1}$, $SS_1$), CF2= (1, $\vec{LS_2}$, $SS_2$), CF3= (1, $\vec{LS_3}$, $SS_3$), CF5= (2, $\vec{LS_2}$ + $\vec{LS_3}$, $SS_2$+ $SS_3$).

Now it descends video 18_9324 from the root to leaf node based on the closest, absorbing criteria resulting in the final CF tree (fig. 3.8)

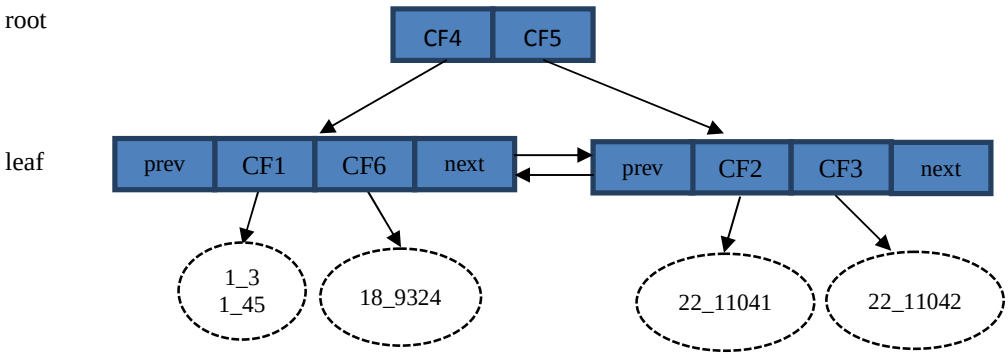



**Figure 3.8. Final initial CF tree result**

The clusters are formed in the leaf. So far, the formed clusters are sensitive to the order of the video insertion. The optional stage could improve the clustering results. Any clustering algorithm such as k-means could be applied to the leaf entries directly to group entries. In paper [19], the authors adapted an agglomerative hierarchical clustering algorithm by applying it directly to the sub

clusters represented by their CF vectors. It uses the accurate distance metric D2 or D4, which can be calculated from the CF vectors, during the whole clustering and has a complexity of O ($N^2$).

### 3.2.3. CURE

CURE [16] can identify both spherical and non-spherical clusters. It chooses a number of well scattered points as representatives of the cluster instead of one point, which is the centroid. To speed up clustering, random sampling and partitioning methods are introduced. The clustering procedures are briefly described as following:

[1] Draw a random sample, S, of the original objects.
[2] Partition sample S into a set of partitions.
[3] Partially cluster each partition.
[4] Eliminate outliers by random sampling. If a cluster grows too slowly, remove it.
[5] Cluster the partial clusters. The representative points falling in each newly formed cluster are “shrinking factor”, α. These points then represent and capture the shape of the cluster.
[6] Mark the data with the corresponding cluster labels.

#### *Illustration*

The same illustration dataset will be used as shownin table 3.1 and table 3.2. To reduce the complexity, the algorithm utilizes two data structures, kd-tree and heap, to store the data information. The kd-tree is used to store the representative data points in every cluster, and the heap stores the cluster in ascending order based on the distance between the cluster and its closest cluster. Since the example dataset is

small, the illustration of how videos are stored will not be shown. It will only go through the key of the cure clustering algorithm.

The selection of representative points is done by choosing the point farthest from the mean as the first scattered point, and then a point farthest from the previous scattered point is chosen until reach the number of c (parameter of the number of representative points). The points are then shrunk toward the mean by a fraction α.

It will present how to partition the dataset into k=3 clusters with c=2 representative points for each cluster. Initially, each video is treated as a separate cluster "u", and computes u.closest for each cluster u. In the example, the result is presented in table 3.11.

***Iteration 1:***

| Cluster u label | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Video member | 1_45 | 1_3 | 22_11041 | 22_11042 | 18_9324 |
| (u.closest, distance) | (2, .4325) | (1, .4325) | (4, .5137) | (3, .5137) | (2, .5354) |

**Table 3.11. Compute closest cluster for each cluster**

After u.closest is computed, It is supposed to insert all clusters into heap in increasing order according to the distance between the cluster u and u.closest. Here it is not using any data structure, only the key concept of the algorithms is presented.

Besides tracking u.closest, it has to maintain the cluster's representative data pointsu.rep as well. In this moment, u.rep of each cluster is only the videos so far have itself, since there is only one video in each cluster in initial pass.

After the data points are initialized into kd-tree T and heap Q, the closest pair of clusters that on the top of Q are removed and merged as a new cluster "w" and the representative data points of the new cluster is recomputed in each iteration until there are only k clusters left. In this case, cluster 1 and 2 are merged as a new cluster 1. The new u.closest will be computed as the least distance between representative points of cluster w to all that of cluster u. Meanwhile, it has to shrink the new representative points of cluster w with factor of α toward the mean by formula p+α*(w.mean-p) where p is the representative point. According to the authors' conclusion a good range value for α, 0.2-0.7, α is set to 0.5. The cluster information is updated as following after first iteration of merge:

Mean of cluster 1 after merging is:
.0384 .0 .0 .0 .0 .0 .0 .0384 .1602 .0 .3268 .0 .0 .0384 .0 .0 .0 .0 .0 .0 .0 .0 .0416 .3557

Applying the shrink formula p+α*(w.mean-p) to all representative points of cluster1, a new value is obtained for 1_45 and 1_3 as shown in table 3.12. The closest cluster information is recomputed and shown in table 3.13.

| | |
|---|---|
| 1_45 | .0192 .0 .0 .0 .0 .0 .0 .0192 .1634 .0 .4134 .0 .0 .0192 .0 .0 .0 .0 .0 .0 .0 .0 .0625 .3029 |
| 1_3 | .0576 .0 .0 .0 .0 .0 .0 .0576 .157 .0 .2403 .0 .0 .0576 .0 .0 .0 .0 .0 .0 .0 .0 .0208 .4086 |

**Table 3.12. The representative points after applying shrunk operation**

| Cluster u label | 1 | 3 | 4 | 5 |
|---|---|---|---|---|
| Representative | 1_45, 1_3 | 22_11041 | 22_11042 | 18_9324 |
| (u.closest, distance) | (5, .5248) | (4, .5137) | (3, .5137) | (1, .5248) |

**Table 3.13. Update the closest cluster information of each cluster**

***Iteration 2:***

The closest clusters are merged in the same fashion by repeating the operations in iteration 1. Cluster 3 and 4 are merged, since they are the closest clusters. After merging, there are three clusters meeting the k=3 condition. Then the iteration stops and the final clusters are formed as 1_45 and 1_3 as one cluster, 22_11041 and 22_11042 as the second cluster, and 18_9324 as the last cluster. The clustering result is the same as the ground truth.

### 3.2.4. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications) [17] is a density based clustering algorithm. Two parameters are required by this algorithm. The parameter Eps ($\varepsilon$) and MinPts are the distance metric and minimum number of points within the specified distance metric (radius) Eps. The cluster is formed by meeting the requirement of MinPts within the radius. To present the idea of DBSCAN, there are a number of definitions involved:

- The neighborhood within a radius $\varepsilon$ of a given object is called the $\varepsilon$-**neighborhood** of the object.
- If the $\varepsilon$-neighborhood of an object contains at least a minimum number, MinPts, of objects, and then the object is called a **core object**.
- The object is called **border point** if it lies on the border of clusters, and has $\varepsilon$-neighborhood less than MinPts
- Given a set of objects, D, it says that an object p is **directly density-reachable** from object q if p is within the $\varepsilon$-neighborhood of q, and q is a core object.
- **Density-reachable** is an extension of directly density-reachable. An object p is density-reachable from object q with respect to $\varepsilon$ and MinPts in a set of

objects, D, if there is a chain of objects $p_1,\ldots,p_n$, where $p_1 = q$ and $p_n = p$ such that $p_i + 1$ is directly density-reachable from $p_i$ with respect to $\varepsilon$ and MinPts, for $1 \leq i \leq n$, $p_i \in D$.

*Algorithm:*

[1] Arbitrary select a point ***p***

[2] Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***

[3] If ***p*** is a core point, a cluster is formed.

[4] If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.

[5] Continue the process until all of the points have been processed

## *Illustration*

| | **1_45** | **1_3** | **22_11041** | **22_11042** | **18_9324** |
|---|---|---|---|---|---|
| **1_45** | 0.0 | | | | |
| **1_3** | 0.4325 | 0.0 | | | |
| **22_11041** | 0.9204 | 0.913 | 0.0 | | |
| **22_11042** | 0.7993 | 0.759 | 0.5137 | 0.0 | |
| **18_9324** | 0.619 | 0.5354 | 0.7489 | 0.6167 | 0.0 |

**Table 3.14. Distance matrix**

The same dataset (table 3.2) is used. The parameters are set *Eps*=0.5 and *MinPts*=1, and video pairwise distance matrix is constructed and shown in table 3.14

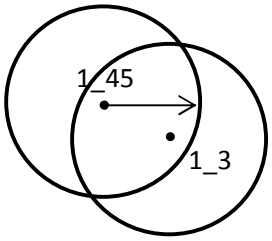



**Figure 3.9. Number of videos within the radius**

It starts to select the point video 1_45 as point p. The $\varepsilon$**-neighborhood** of video 1_45 is 1_3 only, no others wrt to the *Eps*=0.5. There is only one density-reachable point from video 1_45, which is video 1_3. Wrt to *MinPts*=1 and *Eps*=0.5, video 1_45 is a core point (fig 3.9), it retrieves all points density-reachable from it and form the cluster with video 1_45 and 1_3 as membership.

Continuing in this way, it processes next video point. Regarding to the definitions, there are no other core points detected. Hence, with *Eps=0.5 and MinPts=1*, the dataset is classified into 2 clusters, cluster 1 (1_45 and 1_3) and the remaining form the cluster 2. However, the ground truth is video22_11041 and 22_11042 are supposed in one cluster, while video 18_9324 in another cluster.

### 3.2.5. PROCLUS

PROCLUS (PROjectedCLUStering) [12] is a typical dimension-reduction subspace clustering method. It finds the best set of medoids by a hill-climbing process, but generalized to deal with projected clustering. By “hill climbing”, the process successively improves a set of medoids, which serves as the anchor points of the clusters. The Proclus algorithm consists of three phases:

[1] In the initialization phase, it employs the greedy algorithm to select a set of initial medoids that are far apart from each other so as to ensure that each cluster is represented by at least one object in the selected set. A set of points of cardinality a few times larger than k will be picked, where k is the number of clusters expected. After that, it first chooses a random sample of data points proportional to the number of clusters expected, and applies the greedy algorithm to obtain an even smaller final subset for the next phase.

[2] The iteration phase selects a random set of k medoids from this reduced set of medoids, and replaces "bad" medoids with randomly chosen new medoids if the clustering is improved. Two problems are solved in this phase: finding the appropriate set of dimensions for each medoid and forming the clustering corresponding to each medoid. Given the medoids and their associated sets of dimensions, the points are assigned to the medoids using a single pass over the dataset by computing the manhattan segmental distance between the point and the medoid to form clusters.

[3] The refinement phase computes new dimensions for each medoid based on the clusters found, reassigns points to medoids, and removes outliers.

## *Illustration*

The same 5 videos (c.f. table 3.1, 3.2) are chosen as an example and go through the algorithm phase by phase. There are three phases in this algorithm: initialization phase, iteration phase and refinement phase. The Manhattan segmental distance that is defined relative to some set of dimensions *D* is used to compute the distance between any two videos.

***Initialization phase:***

The greedy algorithm is utilized to initialize a set of medoids. There are two parameters for greedy algorithm: a set of sample points *S* and the number of medoids. The size of *S* will be *A* times larger than *k (A·k)*. The number of medoids will be *B·k. B* is a small constant. The final set of medoids, size of *B·k*, is selected from the random sample points set *S* by greedy algorithm as following:

For example there are 5 video points in set *S*= {1_45, 1_3, 22_11041, 22_11042, 18_9324}. Video 1_45 is randomly selected as first medoid $m_1$ from the set *S*, then compute distance between $m_1$ and other videos (table 3.15)

| | 1_3 | 22_11041 | 22_11042 | 18_9324 |
|---|---|---|---|---|
| 1_45 | 0.0368 | 0.0763 | 0.0694 | 0.0782 |

**Table 3.15. Distance of videos to the medoid m1**

The video that is far from the first medoid 1_45 will be chosen as second medoid$m_2$, which is 18_9324 in this case with distance 0.0782. Now it updates the distance table for $m_2$ and the rest videos in sample (table 3.16)

| | 1_3 | 22_11041 | 22_11042 |
|---|---|---|---|
| 18_9324 | 0.0603 | 0.0782 | 0.0748 |

**Table 3.16. Distance of videos to the medoid m2**

According to the greedy algorithm, the third medoid $m_3$ will be 22_11041, since it has the most distance from last medoid $m_2$. Repeating the process in the same fashion until there are *B·k*videos are chosen from sample video set *S* of size *A·k*, where *B* is a small constant and *A* is a larger constant.

It is hard to give the illustration with a large dataset showing *B* and *A* times points/medoids set. However, it has demonstrated how to choose medoids from sample set above. Next, it assumes that medoids set has obtained as*M*= {1_45, 22_11041, 1_3, 22_11042, 18_9324} with size of $B \cdot k$, where sets k=3 to keep consistency with the k in other algorithms illustrations. With the set of *M*medoids, it continues the illustration of iteration phase.

***Iterative phase:***

In iteration phase, it will randomly choose *k* medoids from the *M* set and progressively improve the quality of medoids by iteratively replacing the bad medoids in current set with new medoids from *M* set. Meanwhile, clusters will be formed during the process. The illustration is detailed as following:

As the need of illustration, other videos are added in the small example dataset, 10_4547, 10_4562, 15_6653, 1_110, and 1_120.

Firstly, it randomly selects*k=3*medoids from the set *M as* $M_{current}$={$m_1$:22_11041, $m_2$:18_9324, $m_3$:1_45}.

| | $m_1$:22_11041 | $m_2$:1_45 | $m_3$:18_9324 |
|---|---|---|---|
| 22_11041 | | | |
| 1_45 | 0.0763 | | |
| 18_9324 | 0.0782 | 0.0782 | |

**Table 3.17. Minimum distance between the medoid and other medoids in current medoids set**

Secondly, it has to compute the locality set $L=\{L_1, \ldots, L_k\}$, it sets$k$=3, therefore the set $L=\{L_1, L_2, L_3\}$. Each $L_i$ is a set of video points in sphere centered at corresponding medoid $m_i$ with radius $\delta_i$, $\delta_i$ is the minimum distance between $m_i$ and all other medoids in $M_{current}$, i.e. $\delta_i = \min_{j \neq i} d(m_i, m_j)$. Thus it has $\delta=\{ \delta_1=0.0763, \delta_2=0.0763, \delta_3=0.0782\}$ as shown in table 3.17 according to the distance table 3.18.

| | 1_45 | 1_3 | 22_11041 | 22_11042 | 18_9324 | 10_4547 | 10_4562 | 15_6653 | 1_110 | 1_120 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1_45 | | | | | | | | | | |
| 1_3 | 0.0368 | | | | | | | | | |
| 22_11041 | 0.0763 | 0.0833 | | | | | | | | |
| 22_11042 | 0.0694 | 0.0705 | 0.0416 | | | | | | | |
| 18_9324 | 0.0782 | 0.0603 | 0.0782 | 0.0748 | | | | | | |
| 10_4547 | 0.0645 | 0.0591 | 0.0698 | 0.0685 | 0.0393 | | | | | |
| 10_4562 | 0.0645 | 0.0591 | 0.0698 | 0.0685 | 0.0393 | 0.0 | | | | |
| 15_6653 | 0.054 | 0.0514 | 0.0602 | 0.0656 | 0.0474 | 0.0325 | 0.0325 | | | |
| 1_110 | 0.0267 | 0.0228 | 0.0833 | 0.0714 | 0.0765 | 0.0631 | 0.0631 | 0.0558 | | |
| 1_120 | 0.0694 | 0.0705 | 0.0833 | 0.0625 | 0.0816 | 0.0725 | 0.0725 | 0.0762 | 0.0595 | |

**Table 3.18. Video pairwise distance table**

Assigning points to $L_i$, it is not necessary to cover all video points in the dataset that is some videos may not be assigned to any locality set. Moreover, one video may be assigned to more than one locality, which is fine too. The locality sets are not necessary to be disjoint. With $k$=3medoids set $M_{current}$, cluster set $L$ is produced as in table 3.19 according to the distance table 3.18.

| | |
|---|---|
| $L_1$ , $\delta_1$=0.0763, $m_1$=22_11041 | 22_11042, 10_4547, 10_4562, 15_6653 |
| $L_2$ , $\delta_2$=0.0763, $m_2$=1_45 | 1_3, 22_11042, 10_4547, 10_4562, 1_110, 1_120 |
| $L_3$ , $\delta_3$=0.0782, $m_3$=18_9324 | 1_3, 22_11042, 10_4547, 10_4562, 15_6653, 1_110 |

**Table 3.19. The locality set computed according to video pairwise distance table**

After obtaining the locality, it has to find dimensions for each locality set.

Firstly, it computes the average distance $X_{1j}$ from the points in $L_1$ to medoid $m_1$ along each dimension *j*, and then compute $Y_1 = \frac{\sum_{j=1}^{d} X_{1,j}}{d}$, where d is the total number of dimensions of 24 in this case and obtain the result equal 0.0604. In same fashion, it computes$Y_2$= 0.0553, and $Y_3$ =0.0563.

Secondly, it computes the standard deviation $\sigma_i = \sqrt{\frac{\sum_j \left( X_{i,j} - Y_i \right)^2}{d-1}}$ of the values $X_{i,j}$ and $Z_{i,j} = \frac{X_{i,j} - Y_i}{\sigma_i}$ which indicates how the *j*-dimensional average distance associated with the medoid $m_i$ is related to the average Manhattan segmental distance associated with the same medoid. The computing results are listing in Table 3.20.

| $\sigma_1$ | $\sigma_2$ | $\sigma_3$ |
|---|---|---|
| 0.0985 | 0.09 | 0.0487 |

**Table 3.20. The standard deviation of average distance *X***

| $Z_{3,10}$ | $Z_{3,16}$ | $Z_{3,21}$ | $Z_{3,6}$ | $Z_{3,19}$ | $Z_{3,15}$ | $Z_{3,13}$ | $Z_{2,13}$ | $Z_{2,10}$ | $Z_{3,7}$ |
|---|---|---|---|---|---|---|---|---|---|
| -1.119 | -0.9733 | -0.9342 | -0.8624 | -0.848 | -0.8075 | -0.7741 | -0.6144 | -0.6144 | -0.6057 |
| | | | | | | | | | |
| $Z_{1,10}$ | $Z_{1,13}$ | $Z_{3,23}$ | $Z_{2,3}$ | $Z_{2,4}$ | $Z_{2,19}$ | $Z_{2,12}$ | $Z_{2,16}$ | $Z_{1,12}$ | $Z_{1,19}$ |
| -0.5857 | -0.5857 | -0.5687 | -0.5544 | -0.5544 | -0.5544 | -0.5544 | -0.5544 | -0.5309 | -0.5309 |
| | | | | | | | | | |
| $Z_{1,3}$ | $Z_{2,20}$ | $Z_{2,21}$ | $Z_{1,16}$ | $Z_{1,21}$ | $Z_{2,15}$ | $Z_{3,12}$ | $Z_{3,8}$ | $Z_{2,1}$ | $Z_{1,4}$ |
| -0.5045 | -0.4944 | -0.4944 | -0.4771 | -0.4487 | -0.4355 | -0.4291 | -0.4229 | -0.4122 | -0.3959 |
| | | | | | | | | | |
| $Z_{1,6}$ | $Z_{2,22}$ | $Z_{2,7}$ | $Z_{3,22}$ | $Z_{1,1}$ | $Z_{1,8}$ | $Z_{1,20}$ | $Z_{1,14}$ | $Z_{1,15}$ | $Z_{2,8}$ |
| -0.3949 | -0.3755 | -0.3755 | -0.3675 | -0.3421 | -0.3401 | -0.3137 | -0.3137 | -0.3137 | -0.2922 |
| | | | | | | | | | |
| $Z_{2,14}$ | $Z_{2,5}$ | $Z_{1,22}$ | $Z_{1,7}$ | $Z_{3,18}$ | $Z_{1,11}$ | $Z_{1,18}$ | $Z_{3,14}$ | $Z_{2,6}$ | $Z_{3,3}$ |
| -0.2922 | -0.2555 | -0.2314 | -0.205 | -0.1827 | -0.1502 | -0.1502 | -0.1478 | -0.1244 | -0.0451 |
| | | | | | | | | | |
| $Z_{3,5}$ | $Z_{1,5}$ | $Z_{2,2}$ | $Z_{3,2}$ | $Z_{2,23}$ | $Z_{1,2}$ | $Z_{1,9}$ | $Z_{2,18}$ | $Z_{1,24}$ | $Z_{3,11}$ |
| -0.0431 | -0.0406 | -0.0322 | 0.0369 | 0.1322 | 0.1857 | 0.2121 | 0.26 | 0.336 | 0.4127 |
| | | | | | | | | | |
| $Z_{3,20}$ | $Z_{2,17}$ | $Z_{2,9}$ | $Z_{3,9}$ | $Z_{3,4}$ | $Z_{3,1}$ | $Z_{3,17}$ | $Z_{2,24}$ | $Z_{1,23}$ | $Z_{3,24}$ |
| 0.5359 | 0.73 | 0.9077 | 0.9733 | 1.0657 | 1.5687 | 1.6016 | 1.6611 | 2.336 | 2.8911 |
| | | | | | | | | | |
| $Z_{1,17}$ | | | | | $Z_{2,11}$ | | | | |
| 3.7999 | | | | | 3.8877 | | | | |

**Table 3.21. Sorted Z values in increasing order**

All the *Z* values are sorted in increasing order and listed in Table 3.21. After having sorted Z values for each *L* set, it will select dimensions associated to each set according to the Z value greedily. The original dimensionality is 24, the projected average number of dimensions *l*is set to 12. Then

*kxl=3x12=36*dimensions should be chosen for all 3 sets. In addition, at least two dimensions have to be associated to each set, $Z_{1,10}$, $Z_{1,13}$,$Z_{2,13}$, $Z_{2,10}$, $Z_{3,10}$, $Z_{3,16}$ are selected in the initial stage, since they are the first two least value in each set. Then the rest *kxl*-2*k=30* dimensions will be chosen in increasing order by using the greedy algorithm according to the sorted Table 3.21 as:{ $Z_{3,21}$ , $Z_{3,6}$ , $Z_{3,19}$ , $Z_{3,15}$ , $Z_{3,13}$ , $Z_{3,7}$ , $Z_{3,23}$ , $Z_{2,3}$ , $Z_{2,4}$ , $Z_{2,19}$ , $Z_{2,12}$ , $Z_{2,16}$ , $Z_{1,12}$ , $Z_{1,19}$ , $Z_{1,3}$ , $Z_{2,20}$ , $Z_{2,21}$ , $Z_{1,16}$ ,$Z_{1,21}$,$Z_{2,15}$,$Z_{3,12}$ , $Z_{3,8}$ , $Z_{2,1}$ , $Z_{1,4}$ , $Z_{1,6}$ , $Z_{2,22}$ , $Z_{2,7}$ ,$Z_{3,22}$ ,$Z_{1,1}$ , $Z_{1,8}$}

The associated dimensions to each set is summarized as shown in Table 3.22

| $D_i$ | Associated dimensions to each cluster |
|---|---|
| $D_1$ | 1, 3, 4, 6, 8, 12, 16,19, 21 |
| $D_2$ | 1, 3, 4, 7, 12, 15, 16, 19, 20, 21, 22 |
| $D_3$ | 6, 7, 12, 13, 15, 19, 21, 22, 23 |

**Table 3.22. Associated dimensions to each cluster**

After finding the dimensions associated to each medoid, all video points are assigned to medoids by a single pass over the database. The Manhattan segmental distance between all video points and medoids are computed in corresponding *D* subspace. The video points are assigned to the closest medoid to form the clusters. In the example, it forms the clusters as shown in Table 3.23.

| Cluster id | Cluster members |
|---|---|
| Cluster 1 | 22_11041 |
| Cluster 2 | 1_45, 1_3, 15_6653, 1_110, 1_120 |
| Cluster 3 | 18_9324, 22_11042, 10_4547, 10_4562 |

**Table 3.23. Assign videos to clusters**

After that, it has to evaluate the quality of clusters by computing $Y_{i,j}$, the average distance of points in $C_i$to centroid of $C_i$ along dimension $j$ . The cluster quality is evaluated by computing $\frac{\sum_{i=1}^{k}|C_i|\cdot w_i}{N}$, where $w_i = \frac{\sum_j Y_{i,j}}{|D_i|}$. If the evaluation result is better than the *BestObjective*, The evaluation result is set to *BestObjective*, and $M_{current}$ to$M_{best}$. The medoid of the cluster with least number of video points assigned is defined as a bad medoid. In addition, the medoid of any cluster with less than (N/k)x*minDeviation* points is defined as the bad medoid too. The authors choose *minDeviation=0.1* in their most experiments, while it chooses*minDeviation=0.3* in the demonstration. Thus (N/k)x*minDeviation=(10/3)x0.3=1*. According to the definition of bad medoid, the first medoid22_11041 is bad. The first medoid has to be replaced by another new medoid in medoids set *M*, and repeat processes again until all medoids in *M* are processed. The iterative process will be repeated until meet termination criterion.

***Refinement Phase:***

Getting the best medoids set and clusters from the iterative, the dimensions associated to each medoid will be recomputed and perform the single pass over to assign the points to the closest medoid in related dimensions space.

### 3.2.6. EM

EM (Expectation-Maximization) [11] is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. There are two steps for finding the optimism parameters:

[1] Make an initial guess of the parameter vector: This involves randomly selecting k objects to represent the cluster means or centers (as in k-means partitioning), as well as making guesses for the additional parameters.

[2] Iteratively refine the parameters (or clusters) based on the following two steps:

a) **Expectation Step:** Assign each object to cluster with the probability(c.f. formula 3.1)

$$P(x_i \in C_k) = p(C_k \mid x_i) = \frac{p(C_k)p(x_i \mid C_k)}{p(x_i)} \tag{3.1}$$

Where $p(x_i \mid C_k) = N(m_k, E_k(x_i))$ follows the normal (i.e., Gaussian) distribution around mean $m_k$ with expectation $E_k$.

b) **Maximization Step:** Use the probability estimates from above to re-estimate (or refine) the model parameters. For example (equation 3.2),

$$m_k = \frac{1}{n}\sum_{i=1}^{n}\frac{x_i P(x_i \in C_k)}{\sum_j P(x_i \in C_j)} \tag{3.2}$$

## *Illustration*

In this illustration, the same five videos (c.f. table 4.1) are used as example data. It will initialize parameter k=3 and cluster the data into 3 clusters. The prior probability of each cluster is set equally, which is 1/3 in initial. In addition, the

mean vector and covariance of each cluster gaussian model has to be initialized randomly as well. Three video vectors 1_45, 22_11041, and 18_9324 will be randomly chosen as the mean vectors of 3 clusters respectively, center points in other words. The covariance matrix 24x24 is initialized with diagonal equal 0.1.

***Expectation step:***

In expectation step, it will compute the probability of $i^{th}$ video belonging to $k^{th}$ cluster. By Baysian rule in formula (3.3)

$$P_{k,i} = P(C_k \mid x_i) = \frac{P(x_i \mid C_k) \cdot P(C_k)}{P(x_i)} \tag{3.3}$$

Where $P(X_i)$ is the probability of the $i^{th}$ video, it is computed by theformula (3.4)

$$P_i = \sum_{j=1}^{k} P(C_j) \cdot P(x_i \mid C_j) \tag{3.4}$$

$P(x_i|C_j)$ is computed by the Gaussian distribution with the Mean vector $m_k$ and covariance matrix $\sum_k$ as formula (3.5):

$$P(x_i \mid C_k) = \frac{1}{(2\pi)^{d/2} \left|\sum_k\right|^{1/2}} \cdot \exp\left[-\frac{1}{2}(x_i - \mu_k)^t \Sigma_k^{-1} (x_i - \mu_k)\right] \tag{3.5}$$

Where $|\sum|$ is the determinant of the covariance matrix, $(x_i - \mu_k)^t$ is the transpose vector of $(x_i - \mu_k)$, and $\sum_k^{-1}$ is the inverse covariance matrix of $\sum_k$. After understanding all components in formula, it will compute the probability of all these 5 videos belonging to clusters.

First, it computes the probability of video i given cluster k. In the example, it computes the probability of video i given cluster k, which k=1, 2, 3 and obtain the result in Table 3.24.

| | **Cluster 1** | **Cluster 2** | **Cluster 3** |
|---|---|---|---|
| **1_45** | 0.8224 | 0.0144 | 0.1631 |
| **1_3** | 0.6979 | 0.0184 | 0.2836 |
| **22_11041** | 0.0163 | 0.9275 | 0.0561 |
| **22_11042** | 0.1127 | 0.5693 | 0.3179 |
| **18_9324** | 0.1575 | 0.048 | 0.7943 |

**Table 3.24. The probability of videos belong to clusters**

***Maximization step:***

In maximization, the prior probability, mean vector, and covariance matrix will be updated for each cluster. According to the Table 3.24, it can see that video 1_45 and 1_3 are assigned to cluster 1. Video 22_11041 and 22_11042 are assigned to cluster 2 and cluster 3 has the video 18_9324. In maximization, it will update the

prior probability, mean vector, and covariance matrix for these new formed 3 clusters by using formula 3.1 3.2, 3.5 respectively. After updating, it will repeat the Expectation step and then Maximization step again until it converges. The result is the same to the ground truth.

## 3.3. Analysis

*K-MEANS*

K-MEANS is the most traditional unsupervised clustering method. The algorithm is simple and easy to implement. Its time complexity is O (*nkl*), where n is the dataset size, *k* is the number of clusters, and *l* is the number iterations for the algorithm to converge. The number of clusters *k* has to be pre-defined. The algorithm is not sensitive to the order of dataset input. The metric, measures the distance between two data points, plays an important role deciding the quality of clustering. The most frequent used metric is Euclidean distance; however, other metrics can be employed properly according to the nature of data. Besides, the initial of the cluster center is the key determines the quality of clusters as well. K-MEANS is easy and simple to implement, only one parameter *k* is specified to pre-define the number of clusters to produce.

*DBSCAN*

Dbscan clustering algorithmis density based, thus the formed clusters are separated by the regions with low density. It has to specify 2 parameters radius $\varepsilon$ and MinPts, though it does not require pre-defining the number of expected clusters as Kmeans does. Fig.3.9 shows that video 1_3 is within the radius, which video 1_45 is the core point. The parameter radius $\varepsilon$ will limit the shape of formed clusters to be spherical. To cluster the 5 videos by DBSCAN, if it specifiesMinPts greater than 1,

there will be no more core points, no clusters formed in other words. All videos will be treated as noise. Therefore, it is difficult to find the proper parameters for $\varepsilon$ and MinPts. If the parameters are set improperly, most of videos will be treated as noise. In this algorithm, the order of core points that we picked for forming the clusters is arbitrary. Hence, the formed clusters are sensitive to the order of picking the core points. The core points picked in a different order might result in different clustering results. Regarding to the complexity, it starts with the construction of distance matrix for gathering the neighbors, hence it has the complexity of $O((n^2-n)/2)$. The operation of obtaining neighbors is based on distance (generally the Euclidean distance), thus it may cause the curse of dimensionality issue. When the dataset is large scale, the construction of the distance matrix is rather consuming the main memory. In summary, DBSCAN has the following advantages and disadvantages:

Advantages:

- It is partially insensitive to the order of data in database.
- It does not have to specify the number of clusters in prior.
- Have ability of handling noises.
- Choose the sample points that are well scattered.

Disadvantages:

- Consumes memories for construction of the distance matrix.
- Not be able to find clusters in arbitrary shapes.

### *BIRCH*

Birch algorithm is designed to solve the problems: 1) In the case of large scale datasets, the main memory is limited to fit the dataset in for processing. 2) Thescan

times ofdataset is not focused. 3) I/O costs are very high. It has complexity of *O(n)*, where n is the number of data objects to be clustered in database. Three parameters are required: non-leaf branching contains at most *B* entries; a leaf node contains at most L entries and diameter/radius T of leaf node. CF-tree is the key of the birch algorithm. When the branch/leaf node is violating the factor, the node will be split into 2 and a new node is created as the parent of these 2 branches. The CF node summarizes all the information needed to compute the difference distance metric/diameter in a compact fashion. In the illustration, it can see that, clusters are almost formed in leaf node. Each branch is one sub cluster. Since the ability of how many videos may absorb of leaf node is bounded by the radius T, thus the formed clusters will not be arbitrary shaped. In the example, besides video 22_11041 and 22_11042, others are all correctly grouped in CF-tree. Video 22_11041 and 22_11042 are two near duplicate videos; however, the algorithm did not descend them into the same leaf node entry due to the violation of diameter T caused video 22_11041 is absorbed by leaf entry CF2. Both CF-tree in Birch and distance matrix in Dbscan are trying to find the nearest neighbors. Whilst CF-tree finds nearest neighbors by descend the video over the tree, which is much less memory consumed compared to distance matrix construction. Moreover, the complexity of constructing CF-tree is *O (N)*, which is one scan of the database only. However, searching the nearest neighbors in a CF-tree fashion is sensitive to the order of data scanned. Regarding to the input order issue, the authors proposed phase 3 of global clustering that each sub cluster is represented by the centroid and existing clustering algorithm such as agglomerative hierarchical clustering algorithm is applied directly to the sub clusters. Although phase 3 may help group similar sub clusters, the input order issue inherits from single data object still residents in sub clusters. Hence, it can only say the proposed phase is only solving the input order issue partially. In summary, Birch has the following advantages and disadvantages.

Advantage:

- Low memory consumes.
- Low I/O cost.
- One scan of database only.
- Scalable to large scale dataset.
- CF-tree contains most information needed for computation.

Disadvantage:

- The input order issue is partially solved only.
- 3 parameters have to be re-adjusting frequently to let CF-tree fit into the memory.
- Difficult in forming arbitrary shaped clusters, since it uses the notion of radius/diameter.

### *CURE*

Curehas the worst complexity of $O$ ($n^2 \log n$). 3 parameters are required: 1) the size of sample dataset $S$. 2) the number of clusters $k$. 3) the shrinking factor $\alpha$. Parameters are always required by the clustering algorithm. There has not been such a clustering algorithm that does not require any parameter yet. Therefore, it cannot say that the drawback of this algorithm is that users have to specify parameters. However, more parameters introduced will be a drawback that, it will bring users in difficulty to adjust all parameters for collaborating and producing "best clustering results". The optimization technique may assistance in finding the best parameters combination. However, optimization techniques are always easily trapped in the local optimization and worse results are likely produced. In the CURE algorithm, there is an important operation which cannot be found in any other clustering algorithm, shrinking representative video vector point toward to the

centroid of the cluster by a factor α. The shrinking operation will help eliminate outliers, since representative videos will be pushed closer toward to the centroid and more densely distributed toward to the centroid. In CF-tree of Birch algorithm, video 22_11041 and 22_11042 are located in two clusters respectively, while they could be grouped into the same cluster by Cure merging operation through the manually running the algorithm by using the same 5 videos as illustrated in BIRCH. Because the leaf node of CF-tree is bounded by the radius which may make two videos separated if the boundary is set too low. Although this issue may be solved by later global sub clusters grouping in phase 3 of BIRCH, the videos which do not belong to the sub cluster may be descended to the cluster and cause the deviation of the centroid summarized by the CF vector that, affects the global clustering. However, in Cure clustering, the same issuedoes not exist, since only the nearest neighbor cluster or video will be merged and it does not have to concern of the bounded radius is set too high or too low. In summary, Cure has the following advantages and drawbacks.

Advantages:

- Be able to discover clusters of arbitrary shapes, since it is not use any notion of radius.
- More robust with respect to outliers.
- Be able to cluster a large scale dataset due to clustering on a smaller set of samples, and using data structures heap and kd-tree for fast data access and search.
- Only one scan of the dataset is required.

Disadvantages:

- Users have to specify 3 parameters: 1) the number of clusters *k*. 2) the number of representative data points *c*. 3) the shrinking factor $\alpha$.
- How large the sample shall be drawn will produce the best clustering result?
- The application of clustering on sample dataset results in some clusters missing.

### *PROCLUS*

The Proclus clustering algorithm [16] requires 4 parameters: 1) the number of clusters *k*. 2) constant *B*. 3) factor *A*, which is the number of times larger of *k*. 4) average associated number of dimensions *l*. The algorithm starts with greedily selecting the set of medoids by using hill climbing and end with assigning the data points to the closest medoid form the clusters. The most complicated step is to compute the reduced dimensions associated to the locality set. The computation of projected dimensions for each locality set may solve the "curse of high dimension" problem; however, it will cause the unknown information loss at the same time. In the illustration, videos 22_11041 and 22_11042 are supposed to be grouped in the same cluster, but not. It might be caused by *Manhattan segmental distance* employed instead of Euclidean distance. It also could be the information loss by reducing the associated dimensions to each locality set. In summary, Proclus clustering algorithm has the following advantages and disadvantages.

Advantages:

- The algorithm is good for high dimension, since the clustering happens in subspace.
- Reduce the dimensions.

Disdvantages:

- Have to specify many parameters.
- Hard to determine the average associated number of dimensions that, might produce the best effect.
- Information loss regarding to clustering in the subspace.

### *EM*

EM clustering algorithm is sensitive to the initial configuration and usually gets stuck at local maxima. The algorithm is similar to Kmeans; the difference is that EM uses Gaussian models to initialize each cluster. Two parameters, the mean vector *m* and the covariance of the models, are required to initialize all Gaussian models. Then the probability of each video belonging to each model is computed. Video will be assigned to the cluster of Gaussian model with the highest probability. Then the mean of each model (cluster) will be updated. The formula of video probability computation is shown as equation 1, and equation 2 shows how to update the mean vector for each cluster after all videos are assigned to the corresponding clusters. The probability and mean are computed in the maximization and expectation step respectively. These two steps are repeated until they converge. In summary, it has the following advantages and disadvantages:

Advantages:

- It is sensitive to the initial Gaussian model of each cluster.
- EM is Kmeans-like algorithm, and easy to implement.

Disadvantages:

- Easily gets stuck at local maxima optimization.
- Most of the time, it produces inappropriate overpopulated and under-populate distribution when it is stuck at local optimization.

- Local maxima optimization problem is caused by the algorithm always passing through some high likelihood regions, and skipping the low likelihood regions.

## 3.4. Experiment

The experiment is conducted on PC with Intel(R) Core(TM) 2.67 GHz 2 GB RAM, 32-bit windows7 OS and java programming language is used. The dataset is pre-classified into clusters by these six clustering methods respectively. Then the NDVR with clusters is compared to that of non-clusters based approach. Two parts of experiments are conducted. First, the clustering experiment, the result is shown in Section 3.4.1. This experiment is to study how various clustering techniques partition the dataset to form the clusters with NDVs in the same cluster. Second, having the different clustering results from the first part experiment, the NDVR is conducted with these formed clusters dataset. CC_WEB_VIDEO [44] is used as experiment dataset, which consists of 12860 videos (have more videos than claimed) including 3789 NDVs and 9071 noise videos.

### 3.4.1. Form clusters by various clustering approaches from the literature

To keep the experiments consistence, the parameters of clustering approaches are adjusted to form the number of clusters as close to 1000 as possible. The focus of the clustering is not the accuracy of the clustering approach. The aim is to reduce the dataset accessing time without the retrieval accuracy dropping. According to the small dataset testing, when the number of formed clusters is 6-10 times less than the number of the whole dataset copies, the retrieval time will be reduced around 5 times. Based on the priori knowledge of that, around 1000 clusters are expected to

be formed. Table 3.25 lists the number of clusters formed by applying six clustering approaches.

| | Birch | Cure | Kmeans | Dbscan | Proclus | EM |
|---|---|---|---|---|---|---|
| No. of clusters formed | 1086 | 1000 | 1096 | 950 | 1018 | 999 |

**Table 3.25. Number of clusters formed by different clustering algorithm**

### 3.4.2. NDVR based on clusters formed by different clustering approaches

This section presents NDVR based on the clusters formed by different clustering algorithms. Take Birch clustering as an example. There are 1086 clusters formed by applying Birch clustering algorithm on the 12860 videos dataset. Instead of comparing the query video to 12860 videos in dataset to retrieve the video of interest, the query video is compared to the center (mean vector) of each cluster, 1086 comparison times in other words. The comparison times of one query video is reduced from 12860 times to 1086 times, theoretically, the retrieval time will be 12860/1086 times faster. While the retrieval time is dramatically reduced, the retrieval accuracy is expected to be improved or at least maintained.

Fig. 3.10.shows the precision-recall comparison of NDVR based on clusters formed by the six clustering approaches as well as that of non-clustering based approach. Table 3.27 shows whether there's a significant difference of the retrieval accuracy between various clustering algorithms based retrieval by using t-test statistical testing, where 1 indicates there's a significant difference and 0 indicates there's no significant difference. Their average precision along with the recall is shown in the table 3.26. From the table, dbscan obviously performs the worst

0.3751 only. Kmeans is slightly better compared to no clustering based retrieval, while birch and em are slightly less accurate; however, they are not significant in difference. Through fig. 3.10, the precision-recall curves of birch, em, kmeans and no clustering based are almost superposed. Dbscan precision-recall curve is below that of birch, em, kmeans and no clustering based. It can be concluded that dbscan is not suitable for clustering to enhance NDVR in terms of retrieval accuracy. The same as cure, it has very poor performance as shown in both the table and the figure. Though Proclus has average precision 0.4759, in the precision-recall figure, it can see the performance curve is not stable that, before recall 0.4, its precision is below the other 4 approaches (birch, kmeans, noClusters and em). Thus NDVR based on clusters formed by cure, dbscan and Proclus are poor in performance.

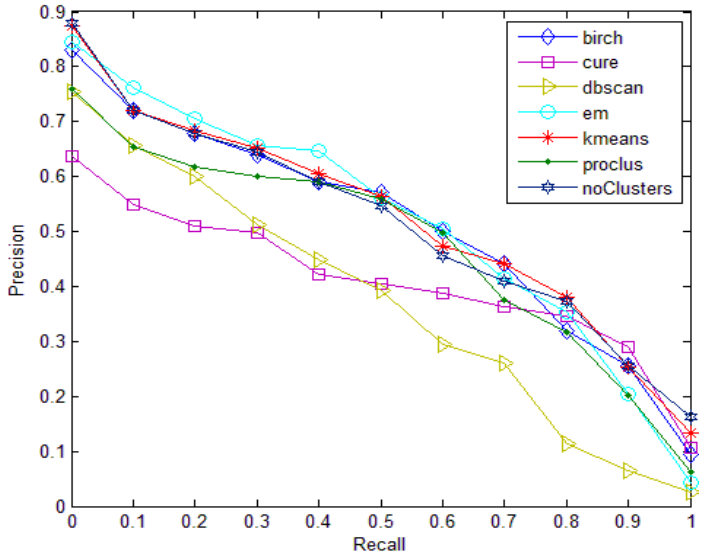



**Figure 3.10. Precision-recall of NDVR based on various clustering algorithms**

| noClusters | Birch | Cure | Dbscan | Kmeans | Em | Proclus |
|---|---|---|---|---|---|---|
| 0.5196 | 0.5133 | 0.4103 | 0.3751 | 0.5254 | 0.5182 | 0.4759 |

**Table 3.26. MAP of precision-recall retrieval based on different clustering**

| | noClusters | Birch | Cure | Dbscan | Kmeans | Em | Proclus |
|---|---|---|---|---|---|---|---|
| noClusters | | | | | | | |
| Birch | 0 | | | | | | |
| Cure | 1 | 1 | | | | | |
| Dbscan | 1 | 1 | 0 | | | | |
| Kmeans | 0 | 0 | 1 | 1 | | | |
| Em | 0 | 0 | 1 | 1 | 0 | | |
| Proclus | 0 | 0 | 0 | 1 | 0 | 0 | |

**Table 3.27. The significant difference relationship of the retrieval accuracy between various clustering-based retrieval**

Fig. 3.11.showsthe time (s) consumed corresponding to different amount of query copies processed. The retrieval time of the six clustering based approach is almost the same, since a similar amount of clusters are formed. However, the same amount of clusters will not produce the same retrieval time. The distribution of amount of video copies in each cluster will affect the retrieval speed too, either a biased distribution or uniform distribution. Fig 3.11.shows that with the number of query videos processed, the retrieval time of the no clustering based increases much faster than that of clustering based. The retrieval time of non-clustering based is

almost two times slower than that of clustering based NDVR when 100 query videos processed. Whilst when 1000 query videos are processed, the retrieval time increases dramatically from 2 times slower to almost 7 times.

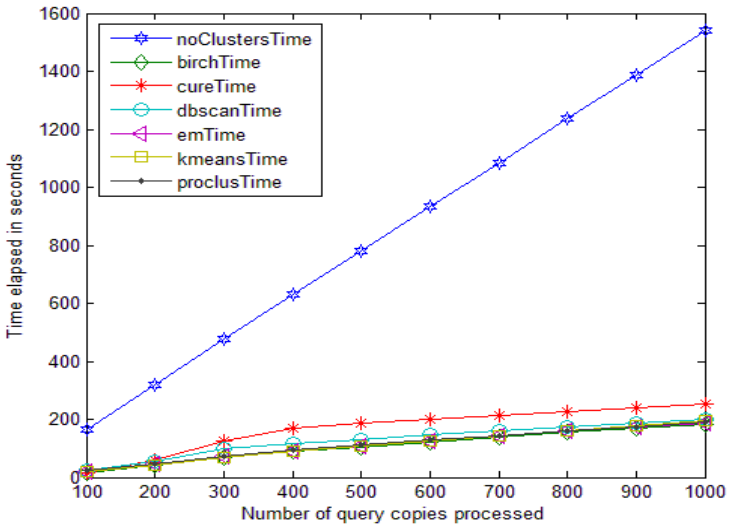



**Figure 3.11. NDVR time at different number of query copies processed**

Finally, itis concluded that NDVR based clusters formed by birch, kmeans and em clustering approaches are promising to enhance NDVR in terms of accuracy and speed, especially the retrieval speed compared to that of non-clustering based approach. The NDVR based on these three clustering methods is much faster compared to that of non-clustering based naïve approach while yields equivalent retrieval accuracy. However, em clustering method is fairly slow, considering that the scale up to millions of videos, kmeans is the best alternative that it is simple in terms of implementation and fast in terms of clustering performance.

## 3.5. Summary

In this chapter, it studies the clustering techniques in literature by illustrating on real video features. The experimental study of the impact of various clustering techniques on NDV retrieval is conducted as well. It used the formed clusters by using various clustering techniques for retrieval evaluation study. Without any doubt, the retrieval speed is increased. However, the empirical results also showus how the retrieval accuracy varies with diverse clustering-based NDV retrieval. This chapter answers the research questions:

*RQ1: Are the clustering algorithms from the literature good for preprocessing NDVs?*

*RQ2: What is the impact of various clustering algorithms from the literature on NDV retrieval?*

And achieves research objectives:

*RO1: To study the clustering algorithms from the literature, and analyze with illustrations.*

*RO2: Experimental study the impact of various clustering algorithms from the literature on NDV retrieval compared to the naïve non-clustering based NDV retrieval. Through the empirical results, it will show the knowledge of which pre-process clustering algorithm is better for NDV retrieval.*

Inthe next chapter, a novel multiple sequence alignment clustering framework will be proposed. The proposed framework caters to the nature of video representation in the form of sequence in video time order.

# CHAPTER4

# *PROPOSED CLUSTERING FRAMEWORK*

Inspired by multiple sequence alignment (MSA) in bioinformatics, MSA will be explored for the content-based video clustering and retrieval. MSA is employed to align the group of DNA sequences and gives the visual view of the similarity between them to determine whether the group of DNA sequences is from the same ancestor. There are three main stages for multiple sequences progressive alignment: first, construct a matrix of all pairwise proximity scores; second, build a guide tree by using neighbor joining according to the proximity matrix created in the first step, progressively joining the two most related sequences until the least related sequences are reached and joined, and all sequences are grouped into the guide tree; third, use the guide tree to further align (a) sequence to sequence, (b) sequence to profile[1], and (c) profile to profile until the root of the tree is reached. In the literature, Kim et al. [26] propose to use the popular gene sequence alignment algorithm in Biology, i.e., BLAST, in detecting near-duplicate images. Based on this idea, they study how various image features and gene sequence generation

[1]More than one video-DNA sequences are aligned as a unit is regarded as the profile

methods (using gene alphabets such as A, C, G, and T in DNA sequences) affect the accuracy and performance of detecting near-duplicate images. Many research papers show video as a string sequence, and apply string sequence alignment as a distance measure[29] [27]. However, MSA has not been exploited for content based near duplicate video analysis yet. This chapter will evaluate whether MSA is feasible for near duplicate videos clustering and the impact on the NDVR based on the clustering results by the MSA clustering framework.

## 4.1. MSA Background

MSA, which is an active research area and has been investigated for decades by bioinformatics researchers, is a sequence alignment algorithm of 3 or more biological sequences, generally protein, DNA, or RNA sequences that are assumed to have an evolutionary relationship by sharing a lineage inherited from a common ancestor. Visual depictions of the alignment (c.f. fig. 4.1.), illustratesmutation events such as point mutations (single amino acid or nucleotide changes) that appear as differing characters in a single alignment column, and insertion or deletion mutations (indels or gaps) that appear as hyphens in one or more of the sequences in the alignment. From the image, it shows how the descendants share the common ancestor. Pairwise alignment is much easier than multiple alignment, it can be aligned by hand easily. However, when there are hundreds or thousands of biological sequences, it becomes difficult and complex or even impossible to align the sequences by hand. Therefore, researchers started exploring multiple sequences alignment methodologies, and trying to reduce its complexity cost for facilitating biologists to conduct the phylogenetic assessment by analyzing the sequences' shared evolutionary origin with the alignment result.

Recently developed systems have advanced the state-of-the art withrespect to accuracy, ability to scale to thousands of proteinsand flexibility in comparing

proteins that do not share the samedomain architecture[46]. Until 2006, the MSA could only be scaled to thousands of proteins and researchers are still trying to exploit variety of ways to reduce the computation complexity cost.

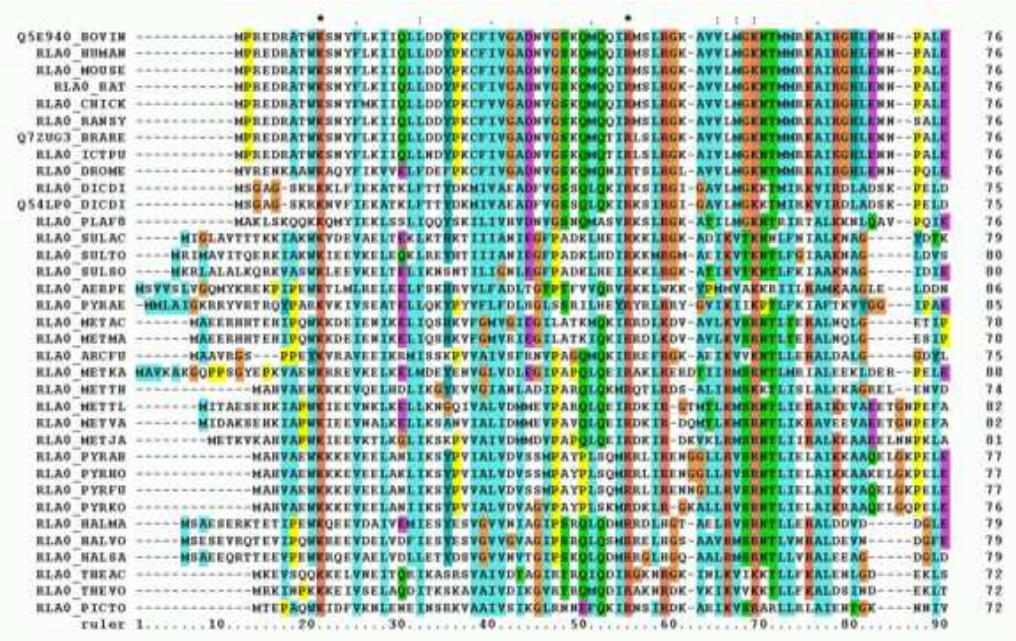


**Figure 4.1.Multiple protein sequence alignment result using ClustalX program [2]**

Approaches such as progressive alignment and iterative methods [34] have been exploited for this purpose, along with the popular dynamic programming algorithm which, however, has been found infeasible for aligning large numbers of sequences due to its exponential complexity [28]. Feng& Doolittle[10] were the first to propose progressive alignment for MSA to address the problem of high computational complexity and real-time processing requirements. More recently, programs such as Clustal series[14] and T-Coffee [42] have been developed that use the progressive alignment principle proposed in [10] to find efficient ways to align multiple biology sequences. Edgar &Batzoglou [46]review most recent MSA programs and conclude that MAFFT [39] [31], MUSCLE [36] [34], T-COFFEE[42]

and PROBCONS [30] are the best current programs in their time. MAFFT and MUSCLE are very similar in design. Both are based on the work which is done by Gotoh in the 1990s that culminated in the PRRN and PRRP programs [15][20]. Although they achieved the best accuracy in their time, they were not widely applied and used due to the slow computation of the program. T-COFFEE is the method based on the prototypical consistency. It is one of the most accurate programs so far. More recently, PROBCONS presented a consistency-based approach that employs a probabilistic model and maximum expected accuracy scoring system [38]. The consistency-based approach shows an advantage in terms of accuracy, but it is very costly to compute. Due to the expensive computational complexity and large memory requirements, PROBCONS and T-COFFEE are practically limited to align around 100 sequences on desktop computers. To address the scalability issue, MAFFT and MUSCLE programs are designed to provide reasonable starting points for general alignment problems with comparable alignment accuracy.There are dozens of programs coming out year by year. However, CrustalW[14] is still the most popular program.

In this thesis, it starts with progressive alignment methodology to solve multiple video sequences alignment problem. The purpose of this is to assess the structure shared by multiple video sequences and make an inference whether they should be grouped into the same cluster, which are from a common ancestor. Progressive alignment methods reduce the problem of aligning all sequences simultaneously to aligning them progressively by aligning the most similar pairs of sequences first, then progressively adding unaligned sequences into aligned profiles until all sequences are aligned. The essence of MSA is using the hierarchical tree clustering algorithm. However, the hierarchical clustering algorithm doesn't really produce clusters, a decision, where to split and form clusters, has to be made. Progressive alignment is a way to split clusters by scoring the multiple sequences

alignment results and assessing the scoring with threshold to form clusters if it meets the criteria.

## 4.2. Proposed video DNA clustering framework

The overview of the framework is proposed in Fig.4.2, illustrating how video collection is partitioned into distinct clusters to further address challenges of accessing a video of interest. The algorithms for each stage process will be detailed in section 3.3. Meanwhile, the clustering flow is demonstrated in algorithm 4.1.

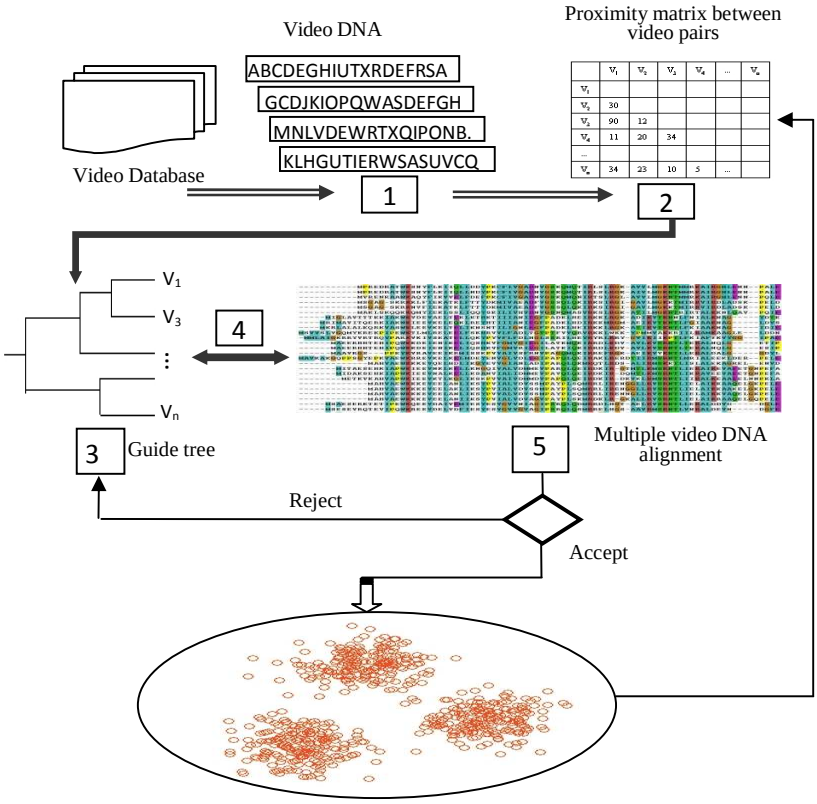



**Figure 4.2.Video DNA clustering framework**

The following will briefly walk through the basic ideas presented in Fig. 4.2.

First, video analysis and features extraction is completed. All videos in the collection are preprocessed through segmentation and keyframe extraction techniques, resulting in a video being characterized by a series of keyframes in chronological order. Representative features for each keyframe are automatically extracted. The entire video is described as a sequence of low-level extracted features, analogous to a DNA or protein sequence, which is called “video-DNA”.

Second,anxn proximity matrix is constructed (where n is the size of the video collection).A proximity measure (using either distance metric/measurement e.g. Euclidean distance/dynamic programming or similarity metric/measurement e.g. cosine similarity/n-gram) is computed for pairs of video sequences and a proximity matrix is constructed.

Third, a guide tree is built which is used to guide the progressive alignment. Based on the proximity matrix, a guide tree is built using a neighbor joining algorithm that: i) creates a profile by joining first the two most related videos to generate the first branching of the guide tree, then ii) progressively joins the less proximate videos are joined into the guide tree in the same manner until all videos are included in the tree in hierarchical groups. Let us note that the guide tree heuristic is only roughly classifying the videos into clusters.

Fourth, the progressive alignment. The refinement of the clustering is done by progressive multiple sequence alignment according to the guide tree (the details are discussed in Section 4.4). Alignment may be between one video and another video (1 to 1 alignment), one video and one video profile (1 to n alignment), or one profile to another profile (m to n alignment).

Fifth, to determine whether the profile is going to a cluster. The video sequence set will be aligned with similar fragments in blocks as shown in Fig.4.2., Stage 5. A score for each profile alignment result is computed and videos in this alignment will be considered members of one cluster if this score is above a given threshold attributed to the cluster.

In order to have high average purity[2] of clusters and DNA-like sequences, we construct the video-DNA as discussed in Section 4.3.1. Each keyframe in a video will be partitioned into 2x2 blocks, the average of intensity of each block is computed and ranked. For a 2x2 partitions, there are 4! Permutations. Each permutation is mapped to the specific alphabet. Integrating the alphabet representation of each keyframe together in a chronological order forms the video-DNA. An n-gram sliding window technique, which has been shown to outperform other state-of-the-art techniques [29], is used to compute the similarity between video pairs as a measure of proximity in the second stage. Normalized Sum-of-Pairs (*SP*) scoring system aims at evaluating the similarity between the profiles resulting from the progressive alignment step.

***Algorithm 4.1: video clustering using MSA***

**INPUT:**Video-NDA, dataset consisting of index video DNA descriptors

**OUTPUT:** Clustered video dataset, each cluster consists of NDV

1- DNA-based Video representation (preprocessing)
2- Proximity Matrix (pair-wise similarity) construction

[2]For the cluster, the video with most number of near-duplicate copies is the dominant cluster, then the number of near-duplicate copies of the dominant is divided by the total number of the cluster is called the purity of the cluster.

3- Guide Tree creation using the Proximity Matrix and a neighbor joining heuristics
4- Progressive alignment according to the Guide Tree
5- Generation of a progressive alignment group proximity score by Sum-of-Pairs scoring and comparison to threshold

**Acceptance:**

a) Retrieve all videos in the current profile as one cluster.
b) Process the un-clustered video DNA representations, repeat from stage 2
c) Continue until all video DNA representations in the tree are classified into clusters

**Rejection:**

Repeat from process 4. Pick a video or profile from the tree according to the degree of proximity and add it to the current alignment. Continue to perform progressive alignment.

---

Algorithm 4.1 gives the detailsof the DNA clustering. In step 1, all videos will be preprocessed into video-DNA offline as shown in Section 4.3.1. In step 2 (c.f. Section 4.3.2), the constructed proximity matrix will be nxn, where n is the number of videos in the database. In step 3 (c.f. Section 4.3.3), the guide tree is built according to the proximity matrix that, the most similar of two members will be selected and joined into the node first and the least similar of two members are joined in last. In step 4 (c.f. 4.3.4), according to order of building the guide tree, the members in the earliest joined two nodes will be selected for alignment to form the profile first, and then the next is selected into the profile for alignment to form the new profile. When the new profile is formed, the step 5 (c.f. 4.3.5) will be introduced to determine whether to send the current profile into the cluster by using

the *SP*scoring system for measurement. Step 2 is repeated when the current profile forms the cluster, otherwise repeat the step 4. The steps repeat in such a manner until all videos are in the corresponding clusters.

# 4.3. Detailed steps

This section will describe the details of the algorithms used at each processing step of the clustering framework introduced in section 2.

## 4.3.1. Video representation for NDV clustering

The CC_WEB_VIDEO [44] dataset is used. It is claimed that there are 12790 videos in dataset. 24 queries were selected to retrieve the most viewed and top favorite videos from YouTube. Each text query was issued to YouTube, Google Video, and Yahoo! Video respectively. The videos were collected in November, 2006. The collected videos have the duration less than 10 minutes. The example videos are shown in fig. 4.3. The shown examples are common in visual content. The clip of near-duplicate durations introduced some rotation transformation, and additional frames/background noise insertion.

Durations of original video clip:

Durations of near-duplicate video clips:

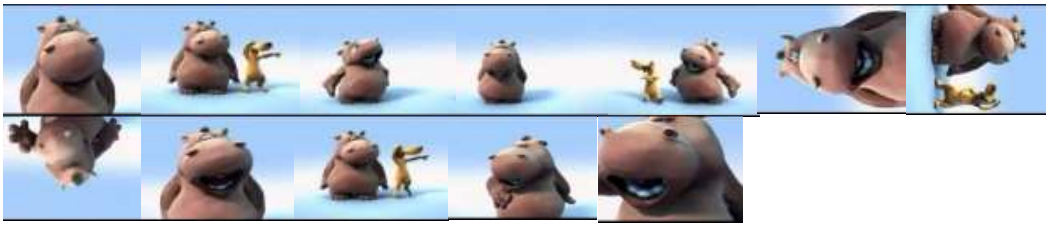


**Figure 4.3.Near-duplicate video examples**

To represent videos as DNA genomes, the idea of the frame macro block spatial ordinal pattern histogram described in [47]will be usedwith slight changes. Fig. 4.4.illustrates how to represent the frame with a single alphabet symbol. As far as ranking is concerned, instead of dividing key frames into 3x2 blocks and using the Y,Cb,Cr color data to compute the block average value, each key frame will be partitioned into 2x2 blocks and use the gray-scale measurements. For 2x2 blocks, we have 4x3x2x1=24 possible permutations of spatial ordinal patterns. For each pattern (permutation), a single alphabet symbol is assigned. In total, 24 alphabets are introduced from A to X. In such a process, each video is a sequence of alphabets character in chronological order, which is calledvideo-DNA.

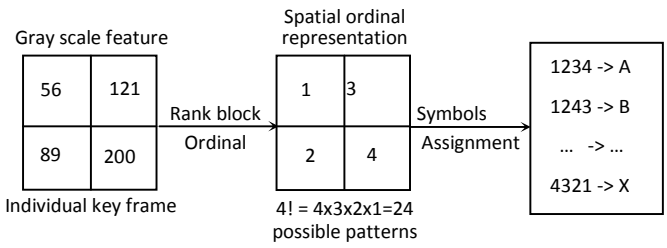



**Figure 4.4.Keyframe pattern extraction processing**

### 4.3.2. Generation of the video proximity matrix

As shown in fig. 4.5., the proximity matrix is symmetrically built up by storing pairwise proximity (similarity measurement is used in the experiment), which is measured by using sliding window based n-gram method.

| | $V_1$ | $V_2$ | $V_3$ | $V_4$ |
|---|---|---|---|---|
| $V_1$ | 0 | | | |
| $V_2$ | 0.27 | 0 | | |
| $V_3$ | 0.15 | 0.43 | 0 | |
| $V_4$ | 0.22 | 0.32 | **0.6** | 0 |

**Figure 4.5.Proximity matrix**

The N-Gram algorithm is used to compute a sliding window based similarity measure. The comparison between two video DNA representations happens only within the window (c.f. fig. 4.6). The window size and step size are given as parameters. Step size is the size to slide the window forward after comparison is done within the window. At each point, the similarity between the two videos within the current window will be calculated and compared to the similarity calculation from the previous window location. The best similarity score will be preserved until the sliding window has moved along the entire video DNA. The detail of the algorithm is shown in algorithm 4.2 [29].

---

***Algorithm 4.2: Sliding-window based similarity technique***

---

**INPUT:** pair-wise DNA video sequence

**OUTPUT:** similarity measure between two videos

1. Apply pair-wise comparison on the n adjacent keyframes each encompassed by a sliding window, count the number of frame matches, and reserve the current offset and the corresponding number of matches.
2. Slide target window S steps forward, repeat step 2.
3. For each complete scan of the target, retrieve the offset with max (number of matched frames) and the query index.
4. Slide query window S steps forward.
5. Repeat from step 2 until complete scan of the query. Assign the best starting index and offset Best (I, O) for the highest number of matches and their offset.
6. Starting from Best (I, O), count the number of equal adjacent pairs until the first non-equal pairs are found and assign a score to the target.


**Figure 4.6.Overview of the sliding window-based similarity technique**

### 4.3.3. Construction of the guide tree

The guide tree is built incrementally following a greedy heuristic neighbor joining algorithm. The selection of two video-DNA descriptors is based on the similarity measurement according to the proximity matrix that two videos with the highest similarity will be chosen, and joined in the same tree node as a profile. In this model, each video DNA is a leaf node of the guide tree. After every join of one

video to one video (1:1) / one video to video profile (1: n) / one video profile to another video profile (m: n), the proximity matrix has to be updated by joining the affected members into the same cell and computing the proximity of the new profile to all of the other videos and video profiles. In such an updating, the size of the proximity is shrinking.

The computation of the proximity of the new profile to all of the other videos or profiles is the average proximity (similarity) of profile member proximity to other videos. For example, there are 4 videos $V_1$, $V_2$, $V_3$ and $V_4$ and assume that the proximity matrix (fig. 4.5.) is computed as described in section 3.3.2. From the proximity matrix, it shows that $V_3$ and $V_4$ are the most similar (value of 0.6). They are then selected and joined to form a branch of the guide tree as shown in fig.4.7. After that, the proximity between the new joined node ($V_3$, $V_4$) and the other videos: $V_1$ and $V_2$have to be updated. The proximity is calculated as the average of the similarities of each member in the joined node to another node, resulting in an updated proximity matrix with 3 nodes: $V_1$, $V_2$ and ($V_3$, $V_4$). The video selection, joining, and matrix updating processes are repeated until all videos in the proximity matrix are joined, resulting in the final guide tree.

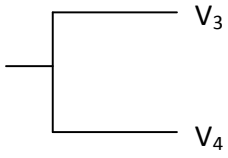



**Figure 4.7.Joining of $V_3$ and $V_4$**

### 4.3.4. Progressive multiple video alignment

In this stage, the selection of videos is according to the guide tree, and aligns them progressively using a dynamic programming technique. The normalized sum of

pairs (*SP*) scoring system is applied to measure the profile alignment at each stage of the process. The scoring result will be used in the cluster formation stage to make a decision on whether videos in the aligned profile shall continue to be considered for the alignment process or be gathered within one cluster.

### *Video sequence selection*

The related videos are selected according to the guide tree constructed in the previous stage. The order of selection follows that of the insertion within the guide tree, which is heuristic. After relevant videos are selected, they are next considered for the progressive alignment process.

### *Progressive alignment*

Rather than aligning all video sequences at one time, the algorithmwill align video sequences progressively. In the very beginning, it always starts with aligning two single video sequences, then followed by selecting another video/profile from guide tree and align with the current formed profile, progressively and repeatedly until it meets the requirement to form the cluster (c.f. Section 4.3.5), and repeat the process of progressively alignment again.

There are three cases to consider for the progressive alignment process: i) *One-to-one*: a single video-DNA sequence is aligned to another single video-DNA sequence; ii) *One-to-profile*: a single video DNA is aligned to many video-DNAs that is aligned (i.e. a profile); iii) *Profile-to-profile*: many aligned video-DNAs are aligned to another many aligned videos-DNAs. In the simplest one-to-one case, two video-DNAs are aligned in the same fashion as two alphabet sequences by using dynamic programming following the optimization alignment path with the delete and insert gap operations.

In the most complex profile-to-profile alignment case, video profiles are treated as single position sequences. Following a dynamic programming technique, an optimization alignment path is searched for. The process is based on introducing *gaps* to align sequences and tracking the optimum path in terms of least cost to fill these gaps (i.e. penalty operations). A dynamic programming table is generated where each row and column correspond to the data of two profiles to be aligned. A similarity score is computed and the optimum path is then tracked. Once a gap is introduced in a profile, it remains open in each sequence at the same position, which follows the policy of "once a gap, always a gap" [33]. The one-to-profile case is a simplification of the profile-to-profile case.

---

***Algorithm 4.3: profile-to-profile alignment***

---

**INPUT:** two profiles, $P^1$ and $P^2$, to be aligned

**OUTPUT:** aligned group of DNA video representations

Let $\mathbf{l_1}$and $\mathbf{l_2}$, the lengths of $P^1$ and $P^2$ respectively (in other words, they are the number of columns); $\mathbf{P^1_i}$ and $\mathbf{P^2_j}$ as $i^{th}$ and $j^{th}$ columns of $P^2$ and $P^2$ respectively; **g** as the gap insertion cost

1. loop i from 1 to $\boldsymbol{l_1}$
2. loop j from 1 to $\boldsymbol{l_2}$
3. dist[i][j] = min(dist[i-1][j-1] + mismatchPenalty(i,j), dist[i][j-1]+**g**, dist[i-1][j]+**g**)
4. if dist[i][j] = dist[i][j-1]+**g**
5. insert gap "-" into $\mathbf{P^1_i}$ of each member video-DNA in $P^1$
6. endif
7. if dist[i][j] = dist[i-1][j]+g

8. insert gap "-" into $\mathbf{P^2_j}$ of each member video- DNA in $P^2$
9. endif
10. endloop
11. endloop

---

Algorithm 4.3 details the process of how to insert gaps to align two video DNA profiles. The demonstration of how to align two profiles by following the optimum path is shown in the dynamic programming table of fig. 4.8., which computes the similarity between pairwise columns in profiles $P^1$ and $P^2$ respectively. Each column, treated as one unit, consists of the edge histogram descriptors and a gap "-" is introduced in the same column of each member of the profile. $f_k^i$ represents the edge histogram descriptor of the $k^{th}$keyframe in video DNA *i.* Fig. 4.8.shows an example of a dynamic programming table where profiles $P^1$ and$P^2$ of lengths 5 and 6 respectively are aligned by following the minimum distance path, which is illustrated by arrows. Following this optimum path in the table, the alignment result of profiles $P^1$ and$P^2$ is produced, forming a new profile of length 7 as shown in fig. 4.9.

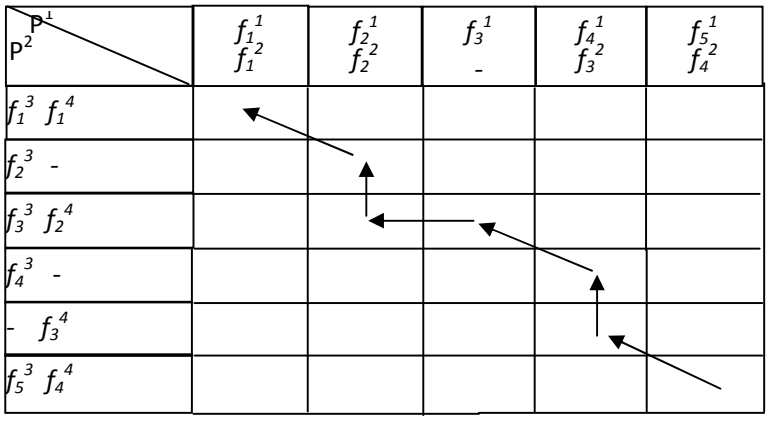



**Figure 4.8. Profile to profile alignment by dynamic programming**

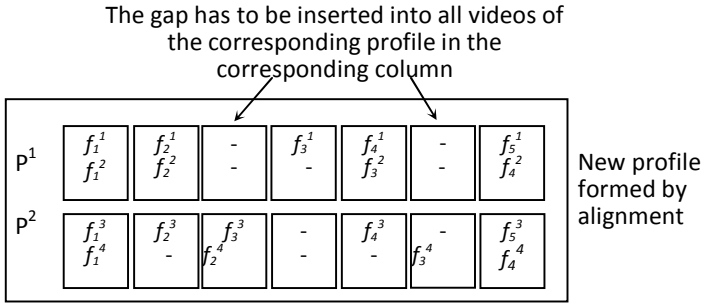



**Figure 4.9. Profile alignment result**

### 4.3.5. Formation of video clusters

Normalized Sum-of-Pairs (SP) scoring is used to evaluate the similarity between the profiles resulting from the progressive alignment step. The final score is computed as the average of each column's normalized score. As an illustration,*n* video-DNA representations is considered in the alignment for which performs the sum up scoring operation, *k* is the number of columns of the aligned profile and *SP(C)*, given in equation (4.1), is the normalized sum of each column score.

$$SP(C) = \frac{1}{k}\sum_{i=1}^{k} SP(c_i) \tag{4.1}$$

Where $SP(c_i)$ is the matching score corresponding to the column position $c_i$for a given video pair. After that, for normalization purposes, the score of each column is divided by the number of comparisons, which is $\frac{nx(n-1)}{2}$.

The scoring result is then compared to a given threshold to assess whether all videos in the corresponding profile will be considered in a cluster. This threshold is especially significant since it determines the similarity of video DNA representations in the same cluster and the number of clusters generated. A high threshold implies high inter-cluster similarity (i.e. *high purity*) and a high number of clusters generated; while a low threshold implies low inter-cluster similarity (i.e. *low purity*) and a small number of formed clusters. Its value is determined experimentally.

## 4.4. Experiment

This section will show the clustering experiments results. The aim of this experiment is to partition the dataset into clusters that NDVs are grouped into the same cluster. By clustering, it achieves the goal of reducing the searching space which is different from the traditional approaches such as indexing techniques.The experiment data is from CC_WEB_VIDEO. There are 12860 videos in CC_WEB_VIDEO dataset, 3789 videos are near duplicate videos, and the rest are noise videos. 5000 videos(all 3789 NDVs and additional 1211 noise videos) are chosen as the experiment dataset.

Two key criteria impact the NDV retrieval performance using a preprocessing step of clustering: i) a low number of clusters is correlated to computational cost effectiveness, as it involves a reduced number of comparisons between the query video and representative cluster videos; ii) higher intra-cluster purity is linked to improved retrieval accuracy, since videos that are the most similar will be clustered together. Hence, the goal is to minimize the number of clusters while maximizing intra-cluster purity.

In the experiment, the clustering threshold is set to 0.3 based on a comparison with other thresholds. With a threshold value of 0.3, 807 clusters are generated and

the average intra-cluster purity is 0.6759. Table 4.1 shows the purity and number of clusters formed for thresholds varying from 0.1 to 0.5 while table 4.2 shows the increment in terms of purity and number of clusters formed corresponding to different threshold intervals. It isnoticed that the increment of purity sharply decreases with increasing threshold values, as well as the increment of the number of formed clusters. The selection of the threshold value and corresponding number of clusters formed is based on optimizing both retrieval accuracy and speed for varying threshold values on a subset composed of 240 videos. The benefit in terms of both retrieval accuracy and speed is apparent when the number of formed clusters is 6 to 10 times smaller than the size of the entire dataset.

| Threshold | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
|---|---|---|---|---|---|
| Purity | 0.5755 | 0.6388 | 0.6759 | 0.7011 | 0.7104 |
| No. of Clusters | 560 | 711 | 807 | 880 | 928 |

**Table 4.1. Purity and number of formed clusters corresponding to different thresholds**

| Threshold Intervals | 0.1-0.2 | 0.2-0.3 | 0.3-0.4 | 0.4-0.5 |
|---|---|---|---|---|
| Increment of Purity | 0.0633 | 0.0371 | 0.0252 | 0.0093 |
| Increment of Number of Formed Clusters | 151 | 96 | 73 | 48 |

**Table 4.2.Increment of purity and number of formed clusters for different threshold intervals**

## 4.5. Summary

This chapter presented an automatic clustering framework based on progressive multiple video DNA sequence alignment for effectively clustering NDV collections. The approach is heuristic and optimized results are produced. The formed 807 NDV clusterswill be used for MSA clustering-based NDV retrieval evaluation in the next chapter. This chapter answers the research questions:

*RQ3: Since video is always represented as a sequence keyframe's descriptor in the order of time, how to develop a clustering technique that caters to the video sequence descriptor?*

And achieves the research objectives:

*RO3: To propose a novel MSA-based clustering algorithm that caters to video sequence signature representation, since the clustering algorithms in literature are good for data points input only that are single vectors with the same length.*

In the next chapter, it will evaluate the MSA-clustering based NDVR performance by using various features.

# CHAPTER 5

# *EVALUATION OF MSA CLUSTERING-BASED NEAR-DUPLICATE VIDEO RETRIEVAL*

This chapter will evaluate the MSA clustering-based NDV retrieval. The evaluation framework is the same as shown in Section 3.1 of Chapter 3. The ordinal feature, global features, and local feature will be evaluated respectively, since various features have different impact on NDV retrieval. The clustering-based retrieval result will be compared to that of non-clustering based naïve NDV retrieval. The purpose of this evaluation is to show that the MSA clustering-based NDV retrieval by using different features is promising compared to that of non-clustering based, since the retrieval speed is greatly reduced while the retrieval accuracy performs equivalent or even better.

## 5.1. Video Features Representation

Color and texture are the most common global features, and SIFT is the most popular local feature. In this chapter, it will evaluate ordinal feature, scalable color, edge histogram and SIFT for clustering-based NDVR.

### 5.1.1. Ordinal feature

Fig. 5.1 shows the process of how to get the 24 bins video histogram by using ordinal information. First, all the keyframes is partitioned into 4 blocks. The average intensity of each block is computed. Then the block is ranked according to the average intensity value. With 4 blocks partition, there are 24 possible permutations, which are also the corresponding bin for the video histogram. The percentage of the keyframes falling into bins is computed as the 24-dimentional video histogram.

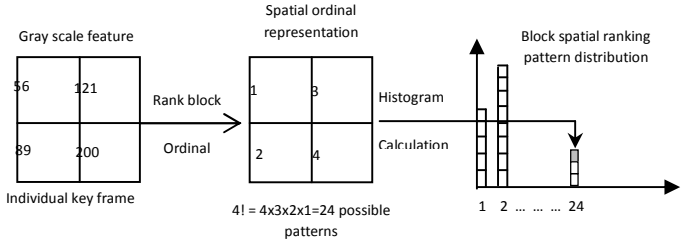



**Figure 5.1. Video spatial pattern histogram processing**

### 5.1.2. Global Features

Color and texture are the most common global visual features. There are various color and texture types [101]. This chapter will evaluate scalable color and edge histogram. Thus, each keyframe could be represented by either a scalable color descriptor or an edge histogram. Therefore, a video is described as a sequence of vector descriptors.

Scalable color is a color histogram that is quantized in the HSV color space and then encoded by Haar transform. City-block distance also known as Manhattan

distance is the best metric to measure the distance between two color histograms [101].

An edge histogram describes the spatial distribution of five types of directional and non-directional edges. The edge histogram outperforms color descriptors. Metric such as Euclidean distance, cosine similarity could be used to measure the distance/similarity between two edge histograms

### 5.1.3. Local Features

Local scale-invariant features (SIFT) [102] was proposed in 1999 by Lowe, D.G. It is still the most popular local feature. Local features outperform global features in terms of accuracy. However, it suffers from expensive computational complexity problem. Accordingly, Sivic and Zisserman [103] proposed text retrieval method by using SIFT, which using k-means to group similar keypoints in the same cluster and mapping keypoints in image/video to closest cluster to form the bag of words (BoW) to describe the image/video. Following this paper, there are many other papers [44] proposed to apply the BoW concept to solve the real-time NDVR in large-scale video collections problem and achieves rather good performance.

BoW concept is employed to process the SIFT feature to video histogram. It randomly selects one representation video for each video cluster. 708 videos are selected. Then it randomly selects 40000 keypoints descriptor from the 708 videos. K-means then applied to cluster 40000 keypoints with k=200. The centroid of each cluster is computed. The size of video histogram equals the size of keypoints clusters which is 200 in this case. Given a video with the SIFT feature extracted, each SIFT point is mapped to the closest cluster. The number of points fall in each cluster bin is computed.

## 5.2. Cluster Representation

The quality of cluster representation is connected to the NDVR performance in terms of accuracy. Two ways are always used to represent the cluster: 1) Select one or more than one videos to represent the cluster. 2) Compute the centroid of the cluster to represent the cluster.

### 5.2.1. Representative Video Selection

In the experiment, for global features (color and textures), it selects one single video to represent the cluster. For each video in the cluster, it computes the number of nearest videos that, meet the required distance/similarity threshold. The video with the greatest number of nearest videos will be selected as the representative video of the cluster.

### 5.2.2. Cluster Centroid Representation

If the video is represented as a histogram (BoW of SIFT and ordinal histogram), the cluster centroid is computed to represent the cluster.

## 5.3. Video Similarity Measurement

This section describesthe similarity measure between two videos. Jaccard Similarity is employed to measure videos described by global features and cosine similarity is applied to score the similarity between two videos represented by local features (video histogram of BoW SIFT) and ordinal feature.

### 5.3.1. Global Features Similarity Measure

Jaccard similarity (Equation 5.1) is used to measure similarity between videos represented by color and texture features. Each keyframe is represented by a color/texture vector, thus the video is described by a sequence of color/texture

vectors. Absolute city-block distance is used to score the distance between two color vectors (scalable color) and cosine similarity is applied to score the similarity between two texture vectors (edge histogram). The measurement will be detailed in algorithm 5.1.

$$Sim(V_1, V_2) = \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|} \quad (5.1)$$

***Algorithm 5.1[3]:Jaccard similarity measure between videos***

INPUT: two sets of video color/texture vectors descriptors $V_1$ and $V_2$ with length $L_1$ and $L_2$

Let variable X be the number of keyframes intersection in two videos, T be the threshold of city-block distance or cosine similarity.

1: LOOP all of keyframes in $V_1$
// score = 100 (if distance);
2: Var score=-1(similarity)
3: LOOP rest of keyframes in $V_2$
// (Var dis=city-block($f_1$, $f_2$) if scalable color)
4: Varsim=cosineSimilarity($f_1$, $f_2$)

[3]The initial for the variable score will be different depending on distance or similarity measurement used. Accordingly, the green color comments are the alternative for the distance measurement used.

// (score>dis && dis<=T)When color distance applied

5: IF (score<sim&&sim>=T)

6: score=sim/dis and record the current keyframe index in $V_2$

7: ENDIF

8: ENDLOOP

// score1!=100 (if distance)

9: IF (score!=-1) {

10: X++; remove the keyframe descriptor at the preserved index

11: ENDIF

12: ENDLOOP

13: get intersection=X; and union= $L_1 + L_2$–intersection

14: apply the jaccard equation (1) and get the similarity

OUTPUT: intersection/union, which is similarity between $V_1$ and $V_2$

---

### 5.3.2. Local and Ordinal Feature Similarity Measure

This section presents the similarity measure between videos by using local and ordinal features. It employs SIFT for the local feature evaluation and processes SIFT to video histogram by using BoW. When an ordinal feature is employed, video is processed into a 24-dimensional video histogram, where each bin is the ranking permutation of average grayscale color of the block in a keyframe. Thus the distance/similarity metric such as Euclidean distance, cosine similarity could be simply applied to measure the distance/similarity between two video histograms. In

this case, cosine similarity metric is employed. Given two video histograms $V^1$and $V^2$, the cosine similarity is measured by the equation (5.2), where k is the length of the video vector.

$$Similarity(V^1, V^2) = \frac{\sum_{i=0}^{k-1} V_i^1 \times V_i^2}{\sqrt{\sum_{i=0}^{k-1} \left(V_i^1\right)^2} \times \sqrt{\sum_{i=0}^{k-1} \left(V_i^2\right)^2}} \quad (5.2)$$

# 5.4. Experimental Results

5000 videos are selected from CC_WEB_VIDEO [104] for the experiment. The 5000 experimental dataset consists of 3789 near duplicate videos and 1211 noise videos. The example videos could be found in [104]. The NDV retrieval evaluation will be conducted on the clusters formed by MSA clustering technique using video-DNA feature (c.f. Chapter 4). Rather than comparing the query video to all videos in database, it only has to compare to representative/centroids of the clusters. By processing this way, the retrieval time is apparently reduced without any doubt. However, the evaluation has to show how the MSA clustering-based retrieval impacts the retrieval accuracy.

Section 5.2.1, 5.2.2 present how to represent clusters efficiently when various video features are applied for MSA clustering-based NDV retrieval.After clusters are represented by the representative videos or centroids, given a query video, instead of comparing to all 5000 videos (non-clustering based naïve NDVR), it only has to compare to 807 cluster representations/centroids (MSA clustering-based

NDVR). It can see that the clustering process physically reduces the size of dataset offline rather than online process by applying indexing structure. The similarity between videos is measured depending on the usage of ordinal, global and local features as described in Section 5.3.1, 5.3.2 and 5.3.3 respectively. Four empirical evaluations are conducted, which compares the performance of clustering based NDVR to that of non-clustering based by using ordinal, color, texture, and SIFT respectively. The comparison is taken under the same criteria: i) same video feature processing (color, texture and SIFT); ii) same similarity measure method with same threshold; iii) the experiment is conducted on a standard PC with an Intel(R) Core(TM) 2.67GHz processor and 2 GB of RAM.The presentation of evaluation follows the information retrieval standard by using precision-recall curve, where precision is computed by formula 5.3 and recall is computed by formula 5.4. The curve is generated by using the official released software trec_eval [113].

$$\Pr ecision = \frac{the\ number\ of\ \ correctly\ retrieved\ \ NDVs}{the\ number\ of\ \ retrieved\ \ videos} \tag{5.3}$$

$$\mathrm{Re} call = \frac{the\ number\ of\ \ correctly\ retrieved\ \ NDVs}{the\ ground\ \ truth\ number\ of\ \ NDVs\ of\ \ the\ query\ video} \tag{5.4}$$

Section 5.4.1 will show the evaluation results of NDVR based on MSA clustering by using ordinal feature. Section 5.4.2 will discuss the evaluation results of NDVR based on clustering by using global features edge histogram and scalable

color. Section 5.4.3 will present the evaluation result of NDVR based on clustering by using local features SIFT compared to that of non-clustering based.

### 5.4.1. MSA Clustering-based NDVR with Ordinal Feature

1000 query videos, which are chosen from CC_WEB_VIDEO dataset with different variations (caption inserted, rotation, frame drops, frame inserted, brightness changes etc and also some noise videos) are submitted for NDV retrieval evaluation. It first compares the retrieval result to that of non-clustering based and as well as n-gram and edit distance methods. Then it compares to NDVR based on the clustering techniques as shown in Chapter 3.

Fig. 5.2compares retrieval accuracy of the MSA clustering-based to the state-of-the-art methods e.g. n-gram and edit distance by showing average precision-recall curve. Fig. 5.3shows the MSA clustering-based retrieval speed in seconds compared to that of non-clustering based corresponding to the number of query copies processed. Although there is no significant difference in average precision-recall accuracy by t-test statistical summary, the proposed clustering based technique is significant more accurate at some points of recall, while the retrieval speed is more than 5 times faster when all 1000 queries are processed. The MAP accuracy of the clustering-based method also performs slightly better than that of non-clustering-based. In spite, from the figure 5.2, the proposed technique the accuracy drops below non-clustering based and n-gram before recall 0.1, after that the proposed technique always performs the best. Among other techniques, edit distance performs the slowest, n-gram and non-clustering based are superimposed in retrieval speed that non-clustering based yields better precision between recall 0.1 and 0.9. Overall, the precision of the MSA clustering based NDVR always performs better than that of edit distance over all recall.

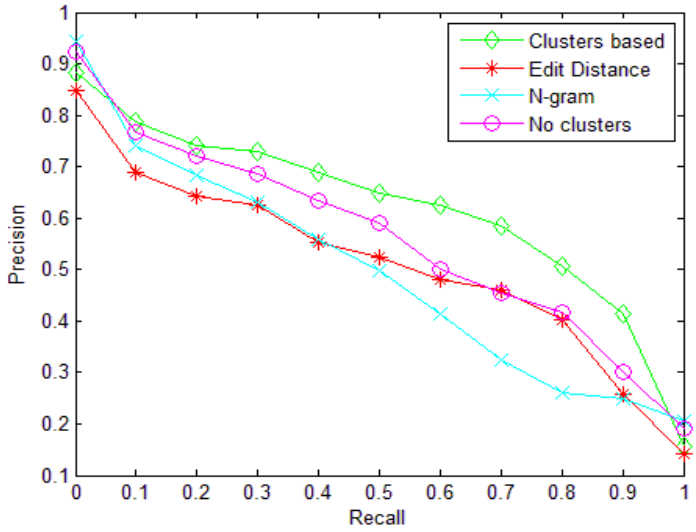



**Figure 5.2.Comparison of retrieval precision-recall by using ordinal feature**

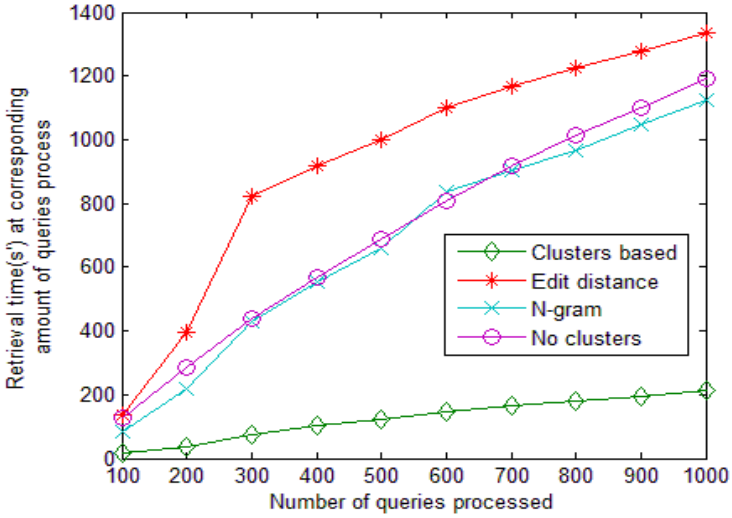



**Figure 5.3.Comparison of retrieval time(s') at corresponding amount of queries processed by using ordinal feature**

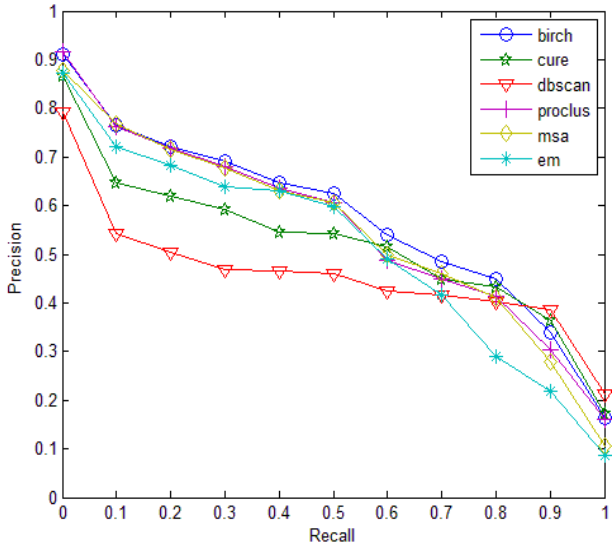



**Figure 5.4. Comparison of MSA clustering-based NDVR tovarious other clustering techniques based NDVR**

Similar to the previous experiment, 1000 queries are considered. For all the compared frameworks, the clustering parameters are adjusted to form a number of clusters sensibly close to that used for MSA based clustering (i.e. 807 clusters). As a consequence, retrieval speed is comparable for all the considered frameworks. However, the video representations used for the 5 clustering algorithms consist of a 24 bin histogram. The corresponding precision-recall curves are shown in fig. 5.4. It can see that the curve of the framework incorporating MSA based clustering is almost superimposed with that of the frameworks implementing clustering based on the Birch and Proclus algorithms. It is tested for the statistical significance of the difference between the average precisions of the proposed framework incorporating MSA based clustering and the frameworks implementing clustering based on the Birch and Proclus algorithms. It is found no significant difference using the t-test:

(t=0.1967, p>0.05, d.f.=19) and (t=-0.0949, p>0.05, d.f.=19) respectively. All three clustering based NDV retrieval frameworks therefore yield equivalent average precision-recall results. Additionally, the framework implementing clustering based on the EM algorithm obviously underperforms the three best performing previous frameworks in terms of precision-recall. In particular, for recall values greater than 0.6, the corresponding precision clearly degenerates. For recall values lower than 0.6 and 0.8 respectively, the curves of the NDV retrieval frameworks incorporating a clustering step based on the Cure and Dbscan algorithms are also clearly below those of the three best performing frameworks. Additionally, the MAPs of all frameworks are listed in table 5.1. Although it did not find any significant statistical difference between the MAP of the proposed framework and that of the frameworks implementing the Birch and Proclus algorithms, the advantage of the proposed MSA based clustering relies on the fact that it is based on alignment, which is more flexible for handling video index signatures whilst other techniques are metric based. To the best of myknowledge, the propose MSA clustering technique is the only that makes use of sequence alignment, which is advantageous for various length of sequence signature based video representations.

| MSA | Birch | Cure | Dbscan | EM | Proclus |
|---|---|---|---|---|---|
| 0.5493 | 0.5713 | 0.5187 | 0.4523 | 0.5055 | 0.5510 |

**Table 5.1. MAP of NDVR based on different clustering algorithms**

## 5.4.2. Clustering-based NDVR with Global Features

This section will show the NDVR performance based on clustering compared to that of non-clustering based by using texture and color respectively. 120 query videos are submitted and the average precision-recall retrieval curves are presented.

Fig.5.5 and fig.5.6 shows the performance by using texture (edge histogram) in terms of retrieval accuracy and speed respectively. Fig.5.5 compares the average precision-recall curve of NDVR based on clustering (with clusters) to that of non-clustering based (no clusters). The figure shows that the clustering based performs much better than non-clustering based before the recall 0.9. The curve is superimposed after the recall 0.9. Overall, clustering-based NDVR has better retrieval accuracy than that of non-clustering based approach. Meanwhile, fig.5.6 shows time consumed corresponding to the number of query copies processed. The clustering based (blue curve) NDVR costs only a little time. The red curve of non-clustering NDVR consumes more time. With the number of query videos that are processed, the time consumed by non-clustering based NDVR climbs much faster than that of clustering based. When 120 videos are processed, clustering based NDVR performs more than 6 times faster than that of non-clustering based. In summary, by using texture feature, clustering based NDVR performs much better than that of non-clustering based in terms both of retrieval accuracy and speed.

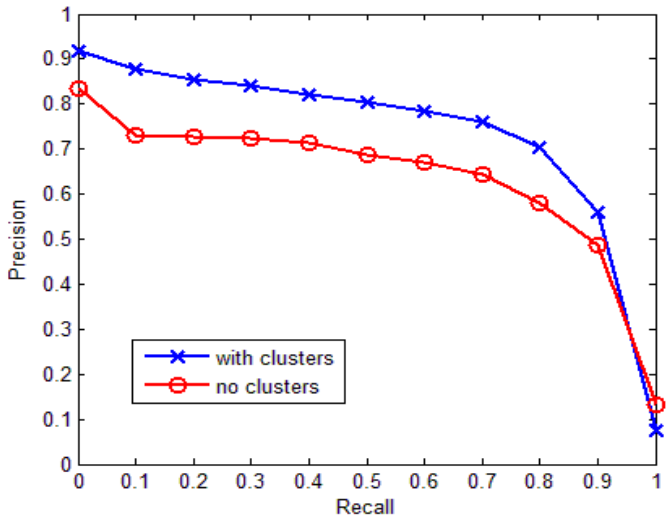



**Figure 5.5.Precision-recall of NDVR by using edge histogram**

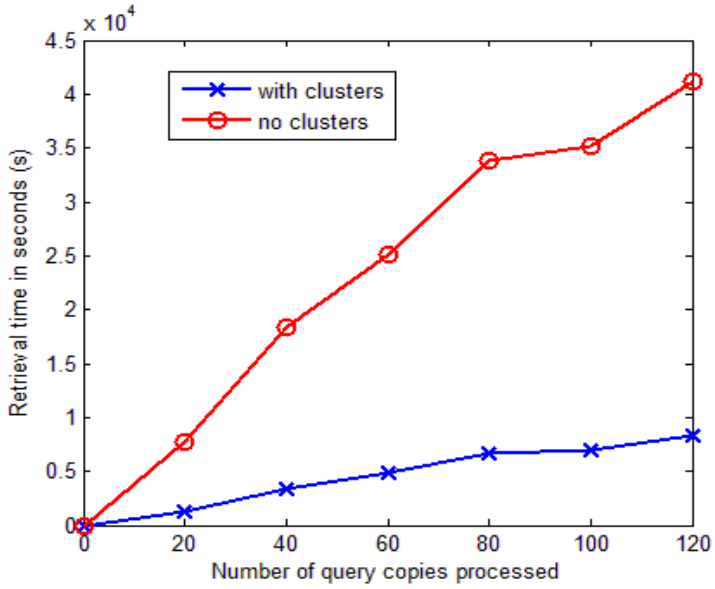



**Figure 5.6.Time consumed corresponding to the number of query copies processedby using edge histogram**

Fig.5.7 and fig.5.8 show NDVR performance by using color (scalable color) in terms of retrieval accuracy and speed respectively. Fig.5.7shows the average NDVR precision-recall curve based on clustering and non-clustering. The blue curve of clustering-based NDVR significantly outperforms the red curve of non-clustering based NDVR before the recall 0.8. However, after the recall 0.8, the two curves are found no significant difference and almost superimposed. Overall, clustering based has significantly better retrieval accuracy. Meanwhile, fig.5.8 presents the retrieval speed, which shows the time consumed corresponding to the number of query videos processed. It can see that NDVR based on clustering (blue curve) consumes a little time only when all 120 query videos are processed and the curve is fairly flat, linearly increasing with the number of query videos increased,

the consumed time climbs much slower than that of non-clustering based approach. By observing the slope of two curves, it can see that the non-clustering based NDVR, the increase of time consumed of processing every single query video is more than 2 times than that of clustering based NDVR. When all 120 query videos are processed, the clustering based NDVR performs around 6 times faster than that of non-clustering based. In summary, by using the color feature, NDVR based on clustering also significantly outperforms non-clustering based in terms of retrieval accuracy and speed.

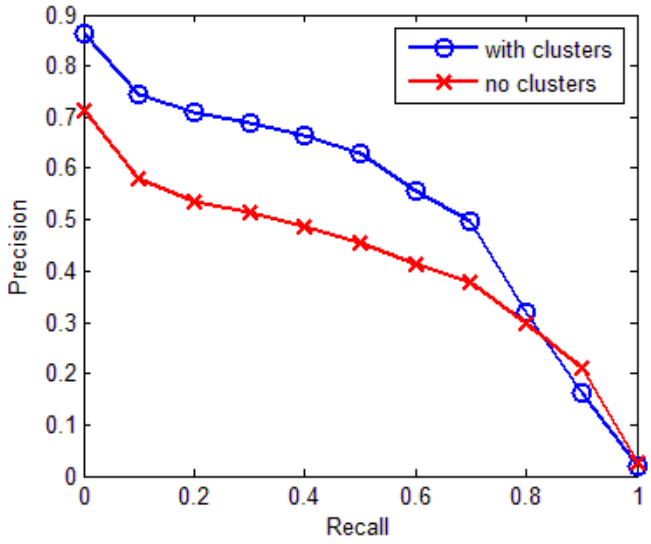



**Figure 5.7.Precision-recall of NDVR by using scalable color**

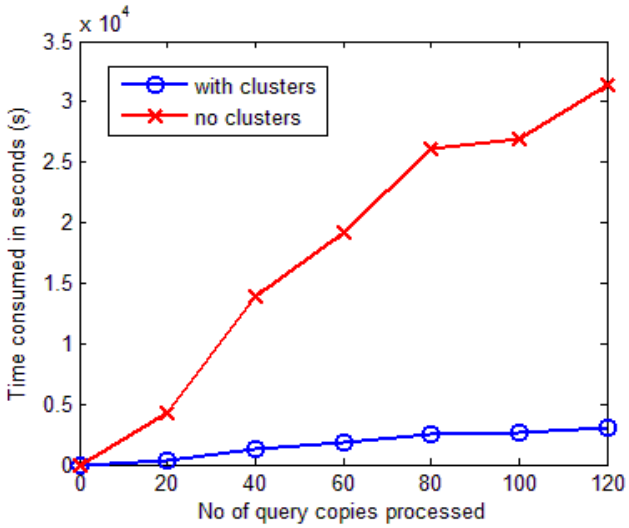



**Figure 5.8.Time consumed corresponding to the number of query copies processedby using scalable color**

### 5.4.3. Clustering-based NDVR with Local Feature

This section will present the NDVR performance in terms of accuracy and speed by using SIFT feature. The same as Section 5.4.2, 120 query videos are submitted for evaluation. The cosine similarity is employed to measure the similarity between videos (c.f. Section 5.3.2).

Fig.5.9 and fig.5.10 show the NDVR performance by using SIFT feature in terms of precision-recall accuracy and speed respectively. As it shows in the fig.5.10, the time consumed by non-clustering based (red) is still greater than that of clustering based (blue), while the accuracy of two precision-recall curves remains at least equivalent. Overall, the two precision-recall curves have no significant difference. However, the retrieval speed is dramatically enhanced by the use of clustering.

When 120 query videos are processed, clustering based NDVR is around 4 times faster than that of non-clustering based.

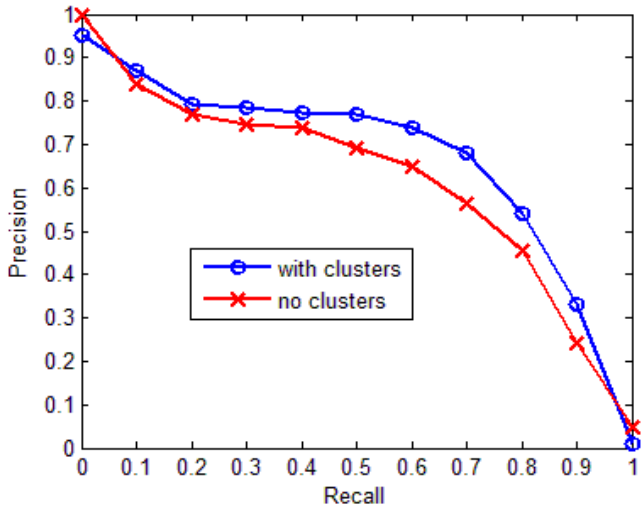



**Figure 5.9.Precision-recall of NDVR by using SIFT**

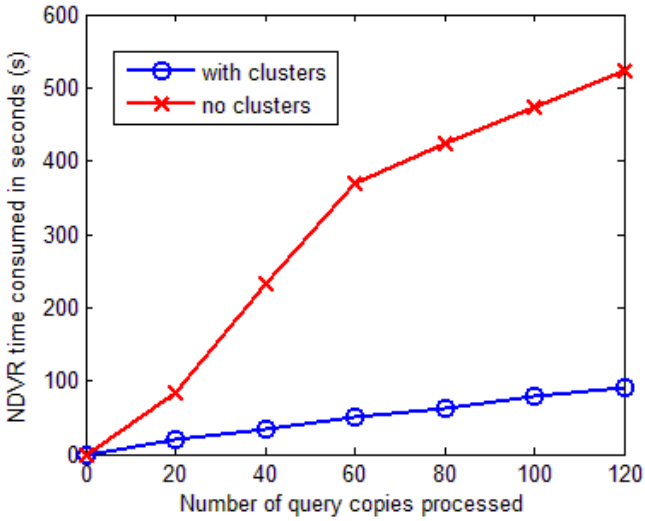



**Figure 5.10.Time consumed corresponding to the number of query copies processed by using SIFT**

## 5.5. Summary

This chapterstudied the effectiveness of NDVR based on MSA clusters compared to that of non-clustering based, and it also compared to the clustering-based retrieval technique to the state-of-the-art technique n-gram and Levenshtein edit distance (dynamic programming) by using ordinal feature. The performance of NDVR with a preprocessing clustering step is better in terms of retrieval cost effectiveness on the one hand and higher in terms of retrieval accuracy in a precision-recall framework on the other hand than other methods.Meanwhile, the retrieval incorporating MSA clustering is also compared against the retrieval that incorporates the clustering techniques. It then also showed that NDVR based on MSA clustering outperforms that of non-clustering based in terms of retrieval accuracy and speed by using popular global and local features. By showing this, itsees that enhancing NDVR through clustering by using popular global and local features is feasible on the dataset. By using clustering techniques to preprocess video dataset offline, it can reduce the size of database offline, which greatly enhance NDVR performance.This chapter answers research questions:

*RQ4: What is the impact of various video features (ordinal, global and local features) impact on MSA clustering-based NDV retrieval.*

*RQ5: What is the performance of MSA clustering-based NDV retrieval?*

And achieves research objectives:

*RO4: To evaluate the MSA clustering based NDV retrieval in terms of retrieval accuracy and speed compared to that of non-clustering based naïve NDV retrieval and also against other clustering methods in literature.*

*RO5: To experimental study how various video feature representations (ordinal, global and local features) impact on MSA-clustering based NDVs retrieval.*

Thenext chapter will give the conclusion and future works.

# CHAPTER 6

# *CONCLUSION AND FUTURE WORK*

## 6.1. Conclusion

In this thesis, in order to explore a new way to improve the NDVR performance in terms of both accuracy and speed, it studies the effectiveness of incorporating the clustering techniques to pre-process dataset into NDVs groups offline and then implement NDVR on the representatives of the clusters to directly reduce the searching space offline rather than online.

First, the impact of different clustering techniques in literature on NDVR was studied. Each clustering method was illustrated and went through the advantages and drawbacks of the techniques. Then, itexperimentally evaluated the NDVR based on clusters formed by using different clustering techniques against that of non-clustering based naïve NDVR. The empirical result presents the feasibility of clustering techniques, which answers research questions:

*RQ1: Are the clustering algorithms from the literature good for preprocessing NDVs?*

*RQ2: What is the impact of various clustering algorithms from the literature on NDV retrieval?*

and achieves research objectives:

*RO1: To study the clustering algorithms from the literature, and analyze with illustrations.*

*RO2: Experimental study the impact of various clustering algorithms from the literature on NDV retrieval compared to the naïve non-clustering based NDV retrieval. Through the empirical results, it will show the knowledge of which pre-process clustering algorithm is better for NDV retrieval.*

Second, to caterto the variable length of video sequence signature representation, a novel clustering framework was proposed based on multiple sequence alignment in Chapter 4, which answers the research question:

*RQ3: Since video is always represented as a sequence keyframe's descriptor in the order of time, how to develop a clustering technique that caters to the video sequence descriptor?*

and achieves the research objective:

*RO3: To propose a novel MSA-based clustering algorithm that caters to video sequence signature representation, since the clustering algorithms are good for data points input only that are single vectors with the same length.*

The approach is heuristic and optimizedthe number of clusters produced to serve the evaluation of MSA clustering-based NDVR which was presented in Chapter 5.

Finally, the evaluation of NDVR incorporating clusters formed by MSA clustering method is conducted. The methods of representing the cluster were introduced. Then a query video is compared to all cluster representatives (clustering-based NDVR) instead of comparing to all videos (non-clustering based NDVR) in the database. Besides the proposed technique was evaluated and compared to non-clustering based retrieval under the same fair criteria, it also

compares to the state-of-the-art methods e.g. dynamic programming and n-gram by using ordinal video features. The result shows clustering-based NDVR is more than 5 times faster than other approaches on one hand and yields better precision at most recall points. Meanwhile, the MSA clustering-based NDVR by using the ordinal feature was also compared against that of theother clustering techniques based. Besides the ordinal feature, the scalable color and edge histogram are chosen as the representatives of the global feature and SIFT as the representative of the local feature to evaluate the impact of the features on NDVR. Since SIFT has exhaustive computation complexity, BoW method is used to process video SIFT into a 200-dimensional histogram. Each bin computes the frequency of the visual word appearing in the video. The experiment resultshows the clustering-based NDVR is much faster while yields significant better precision-recall accuracy by using global features and slightly better MAP accuracy by using local features. This evaluation answers the research questions:

*RQ4: What is the impact of various video features (ordinal, global and local features) impact on MSA clustering-based NDV retrieval.*

*RQ5: What is the performance of MSA clustering-based NDV retrieval?*

andachieve research objectives:

*RO4: To evaluate the MSA clustering based NDV retrieval in terms of retrieval accuracy and speed compared to that of non-clustering based naïve NDV retrieval and also against other clustering methods.*

*RO5: To experimental study how various video feature representations (ordinal, global and local features) impact on MSA-clustering based NDVs retrieval.*

The fullfeasibility study of clustering-based NDVR through different clustering techniques and different features present us prior knowledge of the effectiveness of clustering-based NDVR. In future works, fusion of the clustering and indexing would be studied for both offline and online reduction in the searching space to further accelerate the retrieval speed on the one hand and enforce the retrieval accuracy on the other hand.

Meanwhile, I would like to encourage researchers to consider of pre-processing their dataset by selecting a suitable clustering technique which caters to their scalability and video feature representation requirement before their NDVR operation.

## 6.2. Future Works

### 6.2.1. Fusion of Indexing Structure for Clustering-based NDVR

To achieve the scalability of video retrieval in large-scale database with millions or even billions, the introduction of indexing structure to the framework is the main solution. When videos are processed by clustering methods into clusters, the indexing structure could be introduced to index all the cluster representations instead of indexing all video descriptors.This could assist in further improving the retrieval speed.

As reviewed in chapter 2, many indexing structures such as variety of trees, inverted file, LSH etc. have been studied to index video descriptors. The selection of indexing structure depends on what features used. The inverted file will be introduced when visual words are applied. However, LSH hashing table will be introduced when video is represented in the high-dimensional vector. Multiple hash tables will be incorporated to enhance the retrieval accuracy.

### 6.2.2. Training of Visual Words

Visual words (Bag of Words) have been widely used for image and video retrieval.The basic idea of construction of visual words in video retrieval is to cluster keyframe features by employing clustering algorithms. So far, k-means is widely used for clustering features due to the simplicity and robustness of the algorithm. Then the mean vectors of clusters are computed, and the keyframes of a single video is mapped to the closest cluster by measuring the distance between the keyframe vector and mean vectors of clusters. In such a manner, video is represented by a bag of visual words.

During the construction of the visual words, three issues are important: i) the complexity of visual words construction; ii) how the number of visual words (clusters) affects the retrieval performance; iii) the size of training dataset. These three issues have been addressed and still remain a challenge. However, as far as clustering algorithm concerns, k-means is always used to cluster the sample dataset for visual words construction. Therefore, it will be interesting to study other clustering algorithms and observe how other clustering algorithms impact the visual words based NDV retrieval

### 6.2.3. Video Signature Descriptor

Currently, the most popular video signature descriptor is the BoW, which quantizes frame feature vectors, and then map each keyframe of the videoto the closest visual words which aggregating together to be a BoW describing the video content. Although the technique solves the video retrieval scalability caused by using SIFT features, there are still several uncertain remaining in BoW which was discussed in Section 6.2.2. Thus, considering of improving good performance ordinal feature is an alternative.

### 6.2.4. ImprovementofMultipleSequenceAlignment

Currently, the applied multiple sequence alignment is aligning video feature sequences (shots). It will be interesting to group video shots into scenes, instead of aligning shots, scenes can be aligned locally.

## 6.3. Summary

This chapter summarizes and concludes how the thesis answers the research questions, and achieves the research objectives by experimental study.It concludes that near-duplicate video retrieval based on clustering techniques is feasible and promising.

Besides, it also highlights the future works in several aspects which could be studied and improved. Overall, the chapter gives the general idea of what goals and works in thesis have been achieved, and listed future work trends.

# *APPENDIX A*

## *FORMULAS*

$$Euclidean\ Dis\tan ce: D0 = ((\vec{x}_{0_1} - \vec{x}_{0_2})^2)^{\frac{1}{2}}$$

$$Manhat\tan Dis\tan ce: D1 = |\vec{x}_{0_1} - \vec{x}_{0_2}| = \sum_{i=1}^{d} |\vec{x}_{0_1}^{(i)} - \vec{x}_{0_2}^{(i)}|$$

$$Centroid: \vec{x}_0 = \frac{\sum_{i=1}^{N} \vec{x}_i}{N}$$

$$Radius: R = \left(\frac{\sum_{i=1}^{N} (\vec{x}_i - \vec{x}_0)^2}{N}\right)^{\frac{1}{2}}$$

$$Diameter: D = \left(\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} (\vec{x}_i - \vec{x}_j)^2}{N(N-1)}\right)^{\frac{1}{2}}$$

Average inter-class distance *D2*, intra-class distance *D3* and variance increase distance *D4:*

$$D2 = \left(\frac{\sum_{i=1}^{N_1}\sum_{j=N_1+1}^{N_1+N_2} (\vec{x}_i - \vec{x}_j)^2}{N_1 N_2}\right)^{\frac{1}{2}}$$

$$D3 = \left(\frac{\sum_{i=1}^{N_1+N_2}\sum_{j=1}^{N_1+N_2} (\vec{x}_i - \vec{x}_j)^2}{(N_1 + N_2)(N_1 + N_2 - 1)}\right)^{\frac{1}{2}}$$

$$D4 = \sum_{k=1}^{N_1+N_2} \left(\vec{x}_k - \frac{\sum_{l=1}^{N_1+N_2} \vec{x}_l}{N_1+N_2}\right)^2 - \sum_{i=1}^{N_1} \left(\vec{x}_i - \frac{\sum_{l=1}^{N_1} \vec{x}_l}{N_1}\right)^2 - \sum_{j=N_1+1}^{N_1+N_2} \left(x_j - \frac{\sum_{l=N_1+1}^{N_1+N_2} \vec{x}_l}{N_2}\right)^2$$

# REFERENCES

[1] http://en.wikipedia.org/wiki/Youtube.

[2] http://en.wikipedia.org/wiki/Multiple_sequence_alignment.

[3] http://www.youtube.com.

[4] http://video.yahoo.com.

[5] http://video.google.com.

[6] http://www.mpaa.org/2006 05 03leksumm.pdf

[7] http://www.comscore.com/press/release.asp?press=1678.

[8] http://www.comscore.com/Press_Events/Press_Releases/2008/09/YouTube_Online_Video_Views/%28language%29/eng-US

[9] ComScore,2010.Available:http://www.comscore.com/Press_Events/Press_Releases/2010/6/comScore_Releases_May_2010_U.S._Online_Video_Rankings/%28language%29/eng-US

[10] D. F. Feng and R. F. Doolittle, “Progressive sequence alignment as a prerequisitetto correct phylogenetic trees,” *Journal of molecular evolution*, vol. 25, no. 4, pp. 351–360, 1987.

[11] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions. 1997*. Wiley, New York.

[12] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park, “Fast

algorithms for projected clustering," *ACM SIGMOD Record*, vol. 28, no. 2, pp. 61–72, 1999.

[13] J. Nafeh, *Method and apparatus for classifying patterns of television programs and commercials based on discerning of broadcast audio and video signals*. Google Patents, 1994.

[14] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic acids research*, vol. 22, no. 22, pp. 4673–4680, 1994.

[15] O. Gotoh, "A weighting system and aigorithm for aligning many phylogenetically related sequences," *Computer applications in the biosciences: CABIOS*, vol. 11, no. 5, pp. 543–551, 1995.

[16] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," in *ACM SIGMOD Record*, 1998, vol. 27, pp. 73–84.

[17] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, 1996, vol. 1996, pp. 226–231.

[18] M. M. Yeung and B. Liu, "Efficient matching and clustering of video shots," in *Image Processing, 1995. Proceedings., International Conference on*, 1995, vol. 1, pp. 338–341.

[19] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *ACM SIGMOD Record*, 1996, vol. 25, pp. 103–114.

[20] O. Gotoh and others, “Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments,” *Journal of molecular biology*, vol. 264, no. 4, pp. 823–838, 1996.

[21] J. MacQueen and others, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1967, vol. 1, p. 14.

[22] Xie, Q., Huang, Z., Shen, H.T., Zhou, X., and Pang, C. Quick identification of near-duplicate video sequences with cut signature. *World Wide Web 15*, 3 (2012), 355–382.

[23] X. Cheng and L. T. Chia, “Stratification-based keyframe cliques for removal of near-duplicates in video search results,” in *Proceedings of the international conference on Multimedia information retrieval*, 2010, pp. 313–322.

[24] V. Kobla, D. Doermann, and C. Faloutsos, “Videotrails: Representing and visualizing structure in video sequences,” in *Proceedings of the fifth ACM international conference on Multimedia*, 1997, pp. 335–346.

[25] R. Lienhart, C. Kuhmunch, and W. Effelsberg, “On the detection and recognition of television commercials,” in *Multimedia Computing and Systems’ 97. Proceedings., IEEE International Conference on*, 1997, pp. 509–516.

[26] H. Kim, H. W. Chang, J. Lee, and D. Lee, “Basil: Effective near-duplicate image detection using gene sequence alignment,” *Advances in Information Retrieval*, pp. 229–240, 2010.

[27] M. C. Yeh and K. T. Cheng, “Video copy detection by fast sequence

matching," in *Proceedings of the* ACM *International Conference on Image and Video Retrieval*, 2009, p. 45.

[28] P. H. Wu, T. Thaipanich, and C. C. J. Kuo, "Detecting duplicate video based on camera transitional behavior," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 237–240.

[29] M. Belkhatir& B. Tahayna.. Near-duplicate video detection featuring coupled temporal and visual structures and logical inference based matching. *Inf. Proc. & Man. 48(3)*, 489-501, 2012

[30] C. B. Do, M. S. P. Mahabhashyam, M. Brudno, and S. Batzoglou, "ProbCons: Probabilistic consistency-based multiple sequence alignment," *Genome Research*, vol. 15, no. 2, pp. 330–340, 2005.

[31] K. Katoh, K. Kuma, H. Toh, and T. Miyata, "MAFFT version 5: improvement in accuracy of multiple sequence alignment," *Nucleic acids research*, vol. 33, no. 2, pp. 511–518, 2005.

[32] J. Yuan, L. Y. Duan, Q. Tian, and C. Xu, "Fast and robust short video clip search using an index structure," in *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, 2004, pp. 61–68.

[33] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic acids research*, vol. 32, no. 5, pp. 1792–1797, 2004.

[34] R. Edgar, "MUSCLE: a multiple sequence alignment method with reduced time and space complexity," *BMC bioinformatics*, vol. 5, no. 1, p. 113, 2004.

[35] M. K. Shan and S. Y. Lee, "Content-based video retrieval based on similarity of frame sequence," in *Multi-Media Database Management Systems, 1998.*

*Proceedings. International Workshop on*, 1998, pp. 90–97.

[36] L. Agnihotri, N. Dimitrova, T. McGee, S. Jeannin, D. Schaffer, and J. Nesvadba, “Evolvable visual commercial detector,” in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 2003, vol. 2, p. II–79.

[37] S. A. Berrani, L. Amsaleg, and P. Gros, “Approximate searches: k-neighbors+ precision,” in *Proceedings of the twelfth international conference on Information and knowledge management*, 2003, pp. 24–31.

[38] R. Durbin, *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge UnivPr, 1998.

[39] K. Katoh, K. Misawa, K. Kuma, and T. Miyata, “MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform,” *Nucleic acids research*, vol. 30, no. 14, pp. 3059–3066, 2002.

[40] Y. Li and C. C. J. Kuo, “Detecting commercial breaks in real TV programs based on audiovisual information,” in *Proceedings of SPIE*, 2000, vol. 4210, p. 225.

[41] Y. P. Tan, S. R. Kulkarni, and P. J. Ramadge, “A framework for measuring video similarity and its application to video query by example,” in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, 1999, vol. 2, pp. 106–110.

[42] C. Notredame, D. G. Higgins, J. Heringa, and others, “T-Coffee: A novel method for fast and accurate multiple sequence alignment,” *Journal of molecular biology*, vol. 302, no. 1, pp. 205–218, 2000.

[43] S. Cheung and A. Zakhor, “Efficient video similarity measurement and search,” in *Image Processing, 2000. Proceedings. 2000 International Conference on*, 2000, vol. 1, pp. 85–88.

[44] X. Wu, A. G. Hauptmann, and C. W. Ngo, “Practical elimination of near-duplicates from web video search,” in *Proceedings of the 15th international conference on Multimedia*, 2007, pp. 218–227.

[45] K. Kashino, T. Kurozumi, and H. Murase, “A quick search method for audio and video signals based on histogram pruning,” *Multimedia, IEEE Transactions on*, vol. 5, no. 3, pp. 348–357, 2003.

[46] R. C. Edgar and S. Batzoglou, “Multiple sequence alignment,” *Current opinion in structural biology*, vol. 16, no. 3, pp. 368–373, 2006.

[47] J. Yuan, Q. Tian, and S. Ranganath, “Fast and robust search method for short video clips from large video collection,” *Pattern Recognition*, vol. 3, pp. 866–869, 2004.

[48] S. S. Cheung and A. Zakhor, “Efficient video similarity measurement with video signature,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 1, pp. 59–74, 2003.

[49] T. C. Hoad and J. Zobel, “Fast video matching with signature alignment,” in *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, 2003, pp. 262–269.

[50] Y. Li, J. S. Jin, and X. Zhou, “Matching commercial clips from TV streams using a unique, robust and compact signature,” in *Digital Image Computing: Techniques and Applications, 2005. DICTA’05. Proceedings 2005*, 2005, p. 39–39.

[51] D. N. Bhat and S. K. Nayar, “Ordinal measures for image correspondence,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 4, pp. 415–423, 1998.

[52] A. Hampapur, K. Hyun, and R. Bolle, “Comparison of sequence matching techniques for video copy detection,” in *Conference on Storage and Retrieval for Media Databases*, 2002, pp. 194–201.

[53] X. S. Hua, X. Chen, and H. J. Zhang, “Robust video signature based on ordinal measure,” in *Image Processing, 2004. ICIP’04. 2004 International Conference on*, 2004, vol. 1, pp. 685–688.

[54] S. Paisitkriangkrai, T. Mei, J. Zhang, and X. S. Hua, “Scalable clip-based near-duplicate video detection with ordinal measure,” in *Proceedings of the ACM International Conference on Image and Video Retrieval*, 2010, pp. 121–128.

[55] N. Yazdani and Z. M. Ozsoyoglu, “Sequence matching of images,” in *Scientific and Statistical Database Systems, 1996. Proceedings., Eighth International Conference on*, 1996, pp. 53–62.

[56] C. W. Chang and S. Y. Lee, “Video content representation, indexing, and matching in video information systems,” *Journal of Visual Communication and Image Representation*, vol. 8, no. 2, pp. 107–120, 1997.

[57] D. A. Adjeroh, M. C. Lee, and I. King, “A distance measure for video sequence similarity matching,” in *Multi-Media Database Management Systems, 1998. Proceedings. International Workshop on*, 1998, pp. 72–79.

[58] D. A. Adjeroh, I. King, and M. Lee, “Video sequence similarity matching,” *Multimedia Information Analysis and Retrieval*, pp. 80–95, 1998.

[59] D. DeMenthon and D. Doermann, “Video retrieval using spatio-temporal

descriptors," in Proceedings *of the eleventh ACM international conference on Multimedia*, 2003, pp. 508–517.

[60] P. Muneesawang and L. Guan, "Automatic relevance feedback for video retrieval," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, 2003, vol. 3, p. III–1.

[61] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 1, pp. 127–132, 2005.

[62] F. Wang and C. W. Ngo, "Rushes video summarization by object and event understanding," in *Proceedings of the international workshop on TRECVID video summarization*, 2007, pp. 25–29.

[63] A. Joly, C. Frélicot, and O. Buisson, "Robust content-based video copy identification in a large reference database," *Image and Video Retrieval*, pp. 511–516, 2003.

[64] K. Iwamoto, E. Kasutani, and A. Yamada, "Image signature robust to caption superimposition for video sequence identification," in *Image Processing, 2006 IEEE International Conference on*, 2006, pp. 3185–3188.

[65] O. Chum, J. Philbin, M. Isard, and A. Zisserman, "Scalable near identical image and shot detection," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007, pp. 549–556.

[66] S. S. Cheung and A. Zakhor, "Fast similarity search and clustering of video sequences on the world-wide-web," *Multimedia, IEEE Transactions on*, vol. 7, no. 3, pp. 524–537, 2005.

[67] H. Greenspan, J. Goldberger, and A. Mayer, “A probabilistic framework for spatio-temporal video representation & indexing,” *Computer Vision—ECCV 2002*, pp. 313–317, 2006.

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

[69] J. Zhou and X. P. Zhang, “Automatic identification of digital video based on shot-level sequence matching,” in *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005, pp. 515–518.

[70] X. Naturel and P. Gros, “A fast shot matching strategy for detecting duplicate sequences in a television stream,” in *Proceedings of the 2nd international workshop on Computer vision meets databases*, 2005, pp. 21–27.

[71] J. K. Lee, J. H. Oh, and S. Hwang, “STRG-Index: spatio-temporal region graph indexing for large video databases,” in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005, pp. 718–729.

[72] H. T. Shen, B. C. Ooi, and X. Zhou, “Towards effective indexing for very large video sequence database,” in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005, pp. 730–741.

[73] H. T. Shen, X. Zhou, Z. Huang, and J. Shao, “Statistical summarization of content features for fast near-duplicate video detection,” in *Proceedings of the 15th international conference on Multimedia*, 2007, pp. 164–165.

[74] L. X. Tang, T. Mei, and X. S. Hua, “Near-lossless video summarization,” in *Proceedings of the 17th ACM international conference on Multimedia*, 2009,

pp. 351–360.

[75] S. H. Kim and R. H. Park, “An efficient algorithm for video sequence matching using the modified Hausdorff distance and the directed divergence,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, no. 7, pp. 592–596, 2002.

[76] Y. P. Tan, S. R. Kulkarni, and P. J. Ramadge, “A framework for measuring video similarity and its application to video query by example,” in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, 1999, vol. 2, pp. 106–110.

[77] M. R. Naphade, R. Wang, and T. S. Huang, “Multimodal pattern matching for audio-visual query and retrieval,” in *Proceedings of the Storage and Retrieval for Media Datbases*, 2001, pp. 188–195.

[78] M. Bertini, A. Del Bimbo, and W. Nunziati, “Video clip matching using mpeg-7 descriptors and edit distance,” *Image and video retrieval*, pp. 133–142, 2006.

[79] D. A. Adjeroh, M. C. Lee, and I. King, “A distance measure for video sequences,” *Computer Vision and Image Understanding*, vol. 75, no. 1, pp. 25–45, 1999.

[80] S. J. F. Guimaraes, R. K. R. Coelho, and A. Torres, “Counting of video clip repetitions using a modified bmh algorithm: Preliminary results,” in *Multimedia and Expo, 2006 IEEE International Conference on*, 2006, pp. 1065–1068.

[81] Y. Kim and T. S. Chua, “Retrieval of news video using video sequence matching,” in *Multimedia Modelling Conference, 2005. MMM 2005.*

*Proceedings of the 11th International*, 2005, pp. 68–75.

[82] X. Zhou, and H. T. Shen, "Efficient similarity search by summarization in large video database," in *Proceedings of the eighteenth conference on Australasian database-Volume 63*, 2007, pp. 161–167.

[83] M. Covell, S. Baluja, and M. Fink, "Advertisement detection and replacement using acoustic and visual repetition," in *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, 2006, pp. 461–466.

[84] H. Lu, B. C. Ooi, H. T. Shen, and X. Xue, "Hierarchical indexing structure for efficient similarity search in video retrieval," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 11, pp. 1544–1559, 2006.

[85] L. Shang, L. Yang, F. Wang, K. P. Chan, and X. S. Hua, "Real-time large scale near-duplicate web video retrieval," in *Proceedings of the international conference on Multimedia*, 2010, pp. 531–540.

[86] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," *Recent Advances in Visual Information Systems*, pp. 67–81, 2002.

[87] K. M. Pua, J. M. Gauch, S. E. Gauch, and J. Z. Miadowicz, "Real time repeated video sequence identification," *Computer Vision and Image Understanding*, vol. 93, no. 3, pp. 310–327, 2004.

[88] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval," in *ACM Multimedia*, 2004, vol. 4, p. 5.

[89] X. Yang, P. Xue, and Q. Tian, "Automatically discovering unknown short video repeats," in *Acoustics, Speech and Signal Processing, 2007. ICASSP*

*2007. IEEE International Conference on*, 2007, vol. 1, p. I–1265.

[90] W. L. Zhao, S. Tan, and C. W. Ngo, "Large-scale near-duplicate web video search: Challenge and opportunity," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, 2009, pp. 1624–1627.

[91] G. Lu, "Techniques and data structures for efficient multimedia retrieval based on similarity," *Multimedia, IEEE Transactions on*, vol. 4, no. 3, pp. 372–384, 2002.

[92] S. Park and W. W. Chu, "Similarity-based subsequence search in image sequence databases," *International Journal of Image and Graphics*, vol. 3, no. 1, pp. 31–53, 2003.

[93] T. Can and P. Duygulu, "Searching for repeated video sequences," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, 2007, pp. 207–216.

[94] C. H. Hoi, W. Wang, and M. Lyu, "A novel scheme for video similarity detection," *Image and Video Retrieval*, pp. 541–546, 2003.

[95] J. Zhu, S. C. H. Hoi, M. R. Lyu, and S. Yan, "Near-duplicate keyframe retrieval by nonrigid image matching," in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 41–50.

[96] E. Valle, M. Cord, and S. Philipp-Foliguet, "High-dimensional descriptor indexing for large multimedia databases," in *Proceeding of the 17th ACM conference on Information and knowledge management*, 2008, pp. 739–748.

[97] Z. Huang, H. T. Shen, J. Shao, X. Zhou, and B. Cui, "Bounded coordinate system indexing for real-time video clip search," *ACM Transactions on*

*Information Systems (TOIS)*, vol. 27, no. 3, pp. 1–33, 2009.

[98] L. Shang, L. Yang, F. Wang, K. P. Chan, and X. S. Hua, "Real-time large scale near-duplicate web video retrieval," in *Proceedings of the international conference on Multimedia*, 2010, pp. 531–540.

[99] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 423–432.

[100] X. Zhou, X. Zhou, L. Chen, and A. Bouguettaya, "Efficient subsequence matching over large video databases," *The VLDB Journal*, vol. 21, no. 4, pp. 489–508, 2012.

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

[102] Lowe, D.G. 1999. Object recognition from local scale-invariant features. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* (1999), 1150–1157.

[103] Sivic, J. and Zisserman, A. 2003. Video Google: A text retrieval approach to object matching in videos. (2003).

[104] http://vireo.cs.cityu.edu.hk/webvideo/

[105] H. Min, W. De Neve, and Y. M. Ro, "Towards using semantic features for near-duplicate video detection," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pp. 1364–1369.

[106] Wu, X. et al., 2009. Real-time near-duplicate elimination for web video search with content and context. *Multimedia, IEEE Transactions on*, 11(2),

196–207.

[107] Chum, O., Philbin, J. & Zisserman, A., 2008. Near duplicate image detection: min-hash and tf-idf weighting. In Proceedings *of the British Machine Vision Conference*. p. 4.

[108] Basharat, A., Zhai, Y. & Shah, M., 2008. Content based video matching using spatiotemporal volumes. *Computer Vision and Image Understanding*, 110(3), 360–377.

[109] Cherubini, M., de Oliveira, R. & Oliver, N., 2009. Understanding near-duplicate videos: a user-centric approach. In *Proceedings of the seventeen ACM international conference on Multimedia*. pp. 35–44.

[110] Oliveira, R.D., Cherubini, M. & Oliver, N., 2010. Looking at near-duplicate videos from a human-centric perspective. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 6(3), 1–22.

[111] S. Poullot, O. Buisson, and M. Crucianu, "Scaling content-based video copy detection to very large databases," *Multimedia Tools and Applications*, vol. 47, no. 2, pp. 279–306, 2010.

[112] Y. Cai, L. Yang, W. Ping, F. Wang, T. Mei, X. S. Hua, and S. Li, "Million-scale near-duplicate video retrieval system," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 837–838.

[113] http://trec.nist.gov/trec_eval/.

[114] http://www.comscore.com/Insights/Press_Releases/2012/6/comScore_Releases_May_2012_U.S._Online_Video_Rankings

[115] http://www.comscore.com/Insights/Press_Releases/2013/5/comScore_Releas

es_April_2013_US_Online_Video_Rankings