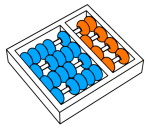
University of Campinas
Institute of Computing



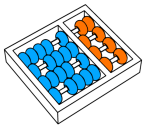
Marcos Vinicius Mussel Cirne

Strategies for Development and Evaluation of Video Summarization Algorithms

Estratégias para Desenvolvimento e Avaliação de Algoritmos de Sumarização de Vídeos

CAMPINAS
2015

University of Campinas
Institute of Computing



Marcos Vinicius Mussel Cirne

Strategies for Development and Evaluation of Video Summarization Algorithms

Estratégias para Desenvolvimento e Avaliação de Algoritmos de Sumarização de Vídeos

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requerements for the degree of Doctor in Computer Science.

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Supervisor/Orientador: Prof. Dr. Hélio Pedrini

Este exemplar corresponde à versão final da Tese defendida por Marcos Vinicius Mussel Cirne e orientada pelo Prof. Dr. Hélio Pedrini.

CAMPINAS
2015

Agência(s) de fomento e nº(s) de processo(s): CNPq, 140781/2013-7

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

C496s

Cirne, Marcos Vinicius Mussel, 1987-
Strategies for development and evaluation of video summarization algorithms / Marcos Vinicius Mussel Cirne. – Campinas, SP : [s.n.], 2015.

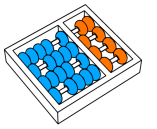
Orientador: Hélio Pedrini.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Sumarização automática. 2. Vídeos - Indexação e resumos. 3. Vídeos - Avaliação. 4. Análise de conteúdo (Comunicação). I. Pedrini, Hélio,1963-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Estratégias para desenvolvimento e avaliação de algoritmos de sumarização de vídeos
Palavras-chave em inglês:
Automatic summarization
Video - Abstracting and indexing
Video - Evaluation
Content analysis (Communication)
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Hélio Pedrini [Orientador]
João Paulo Papa
Paulo André Vechiatto de Miranda
Anderson de Rezende Rocha
José Mario de Martino
Data de defesa: 15-09-2015
Programa de Pós-Graduação: Ciência da Computação

University of Campinas
Institute of Computing



Marcos Vinicius Mussel Cirne

Strategies for Development and Evaluation of Video Summarization Algorithms

Estratégias para Desenvolvimento e Avaliação de Algoritmos de Sumarização de Vídeos

Examining Committee:

- Prof. Dr. Hélio Pedrini
  Instituto de Computação – UNICAMP
- Prof. Dr. João Paulo Papa
  Faculdade de Ciências – UNESP
- Prof. Dr. Paulo André Vechiatto de Miranda
  Instituto de Matemática e Estatística – USP
- Prof. Dr. José Mario de Martino
  Faculdade de Engenharia Elétrica e de Computação – UNICAMP
- Prof. Dr. Anderson de Rezende Rocha
  Instituto de Computação – UNICAMP

A ata da defesa, onde constam as assinaturas dos membros da banca, está arquivada pela Universidade Estadual de Campinas.

Resumo

Sumarização de vídeos é uma área bastante popular que consiste em gerar uma sinopse de um dado vídeo contendo os eventos mais importantes. Ela também é aplicada na análise de grandes quantidades de vídeos digitais, sendo útil na aceleração das tarefas de indexação, busca e recuperação por conteúdo. No entanto, devido ao fato de que existem vários gêneros de conteúdo de vídeos, que incluem esportes documentários, noticiários, programas de TV, entre outros, definir se um evento é ou não importante é um problema em aberto. Várias abordagens têm sido propostas para lidar com esse problema, as quais podem ser ou voltadas para um domínio específico, trabalhando somente com vídeos de um dado gênero, ou de propósito geral, trabalhando com qualquer tipo de vídeo independente do seu gênero. Além disso, a avaliação de métodos de sumarização de vídeos não é uma tarefa trivial, uma vez que isso depende fortemente de fatores subjetivos, normalmente baseados em uma média de opiniões de usuários, para determinar a qualidade dos resumos gerados, o que também se torna um fardo ao comparar um método específico contra outros. Este trabalho discute algumas das estratégias utilizadas para sumarização de vídeos e apresenta três diferentes abordagens que lidam com vídeos de qualquer gênero, juntamente com suas respectivas métricas de avaliação e as evoluções entre essas abordagens. A comparação dos resultados obtidos com outros métodos mostra que foi possível desenvolver um método que não apenas possui uma qualidade superior em relação ao estado da arte mas também é rápido e eficiente, sendo aplicável em ambientes de gerenciamento de vídeos.

Abstract

Video summarization is a very popular field of research which consists of generating a synopsis of a given video containing the most important events. It is also suitable for analyzing large amounts of digital videos, being helpful on speeding up the tasks of indexing, browsing and content-based retrieval. However, due to the fact that there are several genres of video content, which include sports, documentaries, news programs, TV shows, among others, defining whether an event is important or not is an open problem. Many approaches have been proposed to overcome this issue, which can be either domain-specific, working only with videos of a given genre, or general purpose, working with any kind of video regardless of its genre. Furthermore, the evaluation of video summarization methods is not a trivial task, since it strongly depends on subjective factors, commonly based on the average opinion of users, to determine the quality of the produced summaries, which also becomes a burden when comparing a specific method against others. This work discusses some of the strategies used for video summarization and presents three different approaches that deal with videos of any genre, along with their respective evaluation metrics and evolutions between these approaches. The comparison of the obtained results against other methods shows that it was possible to develop a method that not only has a superior quality than the state-of-art but also is very fast and efficient, being applicable in video management environments.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

This chapter describes the problem to be investigated in this work, along with its main motivations, objectives and contributions, as well as the text organization.

1.1 Problem Characterization

With the advent of the newest technologies, it has become much easier and more accessible for people to record high quality videos with their digital cameras, smartphones or tablets. Aside from that, the growth of video hosting websites (including cloud platforms), social networks and video streaming services compels their respective users to upload and share a huge number of videos, leading to an astonishing amount of 300 hours of uploaded video per minute, only on YouTube [6]. Although there is plenty space on servers around the world for the storage all those videos, it demands a great effort for the systems to store them in such a way the tasks of video indexing and retrieval become efficient enough to provide an adequate service for the users.

Fortunately, much research has been done in order to develop techniques that are capable of manipulating these data in an automatic, efficient and accurate way, concerning the issues of searching, browsing, retrieval and content analysis. Among these techniques, we may cite video summarization, which analyzes the content of a given video and creates a snippet that preserves the most important information of this video. However, this process must be conducted in such a way that, by watching the summary, the users must be able to understand at least the most part of the original content without needing to turn to the original video.

Nonetheless, the main problem of developing a video summarization technique is exactly the definition of "important content", since there is a considerable variety of content types (or genres), which include sports, news programs, documentaries, TV series, talk shows and many others, as well as home videos in general. Furthermore, even to humans it is hard to reach a consensus in order to know how good a summary is, because what is relevant to ones may not be to others. Therefore, the elements of video content that must be analyzed or detected to produce a summary must be chosen very wisely, even though the possibilities for this purpose are immense.

Many video summarization techniques have been developed throughout the years, with

the goal of generating, on average, summaries that reflect a "common sense", i.e., which encompass the contents that most of humans would select for a summary. Some of them focus on a specific genre, which makes it easier to define the criteria for choosing the most important events of all videos from that genre. Usually, this type of approach produces very accurate results, but in expense of the genre constraint. On the other hand, there are also approaches that work with any kind of content, but the results are less accurate and demand more generic ways to describe the features that will be used for the summarization process.

Moreover, not only the summaries but also the video summarization methods are difficult to be evaluated among themselves. This happens because each method generally uses their own set of videos for tests, along with a specific evaluation metric, which can be done either by an average opinion obtained from a group of users or by means of an objective metric that compares the produced summaries against others, checking what important contents were included and which ones were left out and, finally, computing a score from these statistics.

This thesis investigates the principles of video summarization and the most common strategies used in the stages of the summarization process, as well as the evaluation metrics and databases used to validate a method. Then, it describes three different approaches:

- **Spectral Clustering:** which performs a special grouping of the frames of a video, selecting one representative frame for each group [31];
- **Image Quality Assessment:** which focuses on defining an objective method for image analysis that considers the way that humans perceive images [32];
- **Color Co-Occurrence Matrices:** which proposes a representation of video frames that yields a more robust approach to generating video summaries in terms of quality;

All these approaches are then compared among themselves and other methods available in the literature, followed by a general discussion about the achieved results.

1.2 Objectives and Contributions

The main objectives of this thesis are as follows:

- Proposal of different strategies for the video summarization problem with their respective advantages, drawbacks, as well as evaluation methods, results and the evolutions among those strategies;
- Comparative analysis of the results obtained by each strategy against several methods of the literature.

The main contributions of this work include:

- Development of a video summarization method that works with any video genre, also having a superior quality in relation to similar methods and being applicable to frameworks that deal with video indexing, retrieval and browsing;
- Increasing of a database of summaries with the produced results so that other methods can make use of them for their own comparisons.

1.3 Text Organization

This thesis is organized as follows: Chapter 2 concerns about video summarization and related concepts, along with a review of some methods of the literature; Chapter 3 describes a proposed video summarization method that uses spectral clustering; Chapter 4 presents another video summarization approach, which uses image quality assessment metrics; Chapter 5 describes the most recent approach proposed, which makes use of color co-occurrence matrices; Chapter 6 concludes this thesis, as well as general perspectives for future work.

Chapter 2

Literature Review

This chapter reviews the theoretical concepts related to the research topic, describing the main approaches to the video summarization research field.

2.1 Related Concepts

This section describes the general concepts about video summarization and shot boundary detection, as well as how the results are evaluated and the main databases used for tests.

2.1.1 Video Summarization

Video summarization [91, 124] consists of deriving a short version from a given video, preserving as much relevant information as possible, such that the users can grasp the message transmitted by the original video. The generated summaries can be integrated into many applications, such as interactive searching and browsing systems, making both management and access to video content more accurate.

When deciding what to look for in a specific video to generate a summary, one may think about specific events that frequently occur on videos of the same content, such as goals or attack situations in soccer games, climax scenes in movies or even an irrelevant content that appears between parts of a TV show, such as advertisings. These elements are known as high-level features, which is a semantic representation of a content that happens on a given instant of the video and refers to subjective aspects that usually vary according to one's interpretation. Other elements that fit into this category are: time, space, objects and human actions. The other possibility is to observe details of the video frames which do not represent directly a semantic element, but are useful for analyzing repeatable patterns or how an object moves in a sequence of frames. Those are called low-level features, and they include: color histograms, texture, motion, audio, subtitles, etc.

According to Taskiran et al. [123], video contents can be classified into two ways:

- *Event-Based Content*: videos that belong to this class contain story units that can be easily identified, forming a sequence of different events, or a sequence of events and non-events. Once the events of interest are well defined, event detection

techniques based on a knowledge domain can be used for the summarization process. In other words, it is necessary to develop a specific application for each sort of content, along with their respective event detection rules, which is a disadvantage. On the other hand, the generated summaries are more trustable than the ones provided by generic summarization algorithms (which work with many video genres). Examples that fit in this category include: talk shows, news programs [71, 120] and sports [16, 33, 41, 45, 81].

- *Uniformly Informative Content*: concerns about every kind of content that cannot be easily split into series of events, as it happens in the aforementioned lcass. Moreover, every part of the video can have the same importance to a user. The summarization of videos that belong to this class is performed from more generic algorithms, intending to eliminate the redundancies of a video sequence, which is done by grouping similar parts of this sequence [9, 48, 55, 57]. Some examples include: documentaries [43, 141], presentations [59], TV shows [137] and home videos [83, 106].

Video summarization techniques can be divided into *static* and *dynamic*. In the first category, the summary is generated as a collection of still images denominated *keyframes* [24, 30, 93, 105], that represent the content of a video in the form of a storyboard [17, 98, 125]. The advantage of this approach is its simplicity and efficiency, usually being free of redundancies, but it may not preserve the temporal order of the selected keyframes. In the second category, many segments of the video are chosen, which are then organized such that the temporal order of the video is preserved [27, 107, 147, 148]. Dynamic summarization has the main advantage of generating summaries which a higher richness of details, but it is computationally more expensive than static summarization approaches, besides the possible generation of redundancies.

There is still a particular category of summarization that makes use of a special kind of video, known as *rush* [99]. Rush videos correspond to the raw material (without editions) used to produce a video [35], such as in movie previews, and have a particular structure, containing redundant segments and useless frames such as color palettes, monochromatic frames and clapboards. Different approaches to rush video summarization have been proposed both for static and dynamic summarization [12, 23, 76, 103], not only to eliminate redundant and useless frames but also to generate good quality summaries. The main advantage of dealing with rush videos is the availability of a specific database (TRECVID [4]) that include, for each original video, a short annotation of the most important scenes that must make part of a good summary, thus constituting a basis for a standard evaluation metric [99]. However, rush video summarization techniques are limited to this type of video, once they have a pre-defined structure.

2.1.2 Shot Boundary Detection

A digital video can be defined as a collection of images that have the same dimensions, being grouped according to a temporal sequence [118]. Each of these images is known as *frame*, which corresponds to the smallest structural unit of a video, representing a photograph captured by a camera in a given time instant of the video. On the other

hand, the frames can be grouped into *shots*, which are sequences of frames captured in a contiguous way and that represent a continuous action in time or in space [84]. Finally, a group of shots that are semantically correlated constitute a *scene*. Figure 2.1 shows a scheme of the general structure of a video.

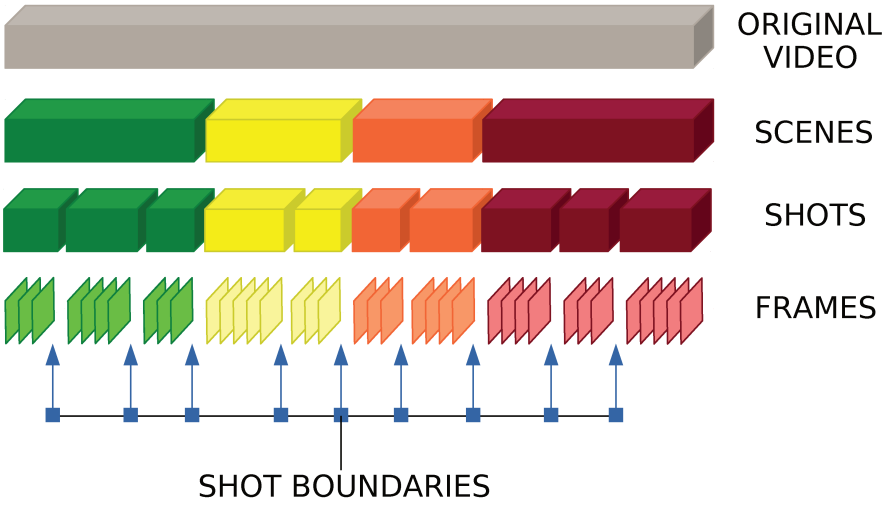


Figure 2.1: General scheme of the structure of a digital video.

Shot boundary detection (SBD) [100, 119] is a fundamental step in video analysis, being applicable not only in video summarization but also in other tasks such as video indexing, browsing and retrieval. Its purpose is to detect and classify transitions between adjacent frames of a video, based on the identification of dissimilarities in frame contents. Shot boundaries can be classified in two different types:

- *Abrupt Transitions or Hard Cuts*: occur when there is a sudden change of content between a pair of consecutive frames from different shots. In other words, given a video of N frames and K shots, an abrupt transition is defined such a way that frame n belongs to shot k and frame $n+1$ belongs to shot $k+1$, where $n \in \{1, 2, ..., N-1\}$ and $k \in \{1, 2, ..., K-1\}$.
- *Gradual Transitions or Soft Cuts*: occur when the content of a shot gradually disappears at the same time that the following shot appears. They are more difficult to be detected because there is little change in the visual content between consecutive frames and the "size" of the transitions (i.e., number of frames that they last) is variable. Furthermore, there are different types of gradual transitions depending on how the visual content changes, which include wipes, dissolves and fade-ins / fade-outs.

Most of the SBD algorithms rely on comparisons of color information of video frames, which are done from a specific metric that calculates the distance (dissimilarity) between two different frames. To detect hard cuts, for instance, if the distance between two consecutive frames is above a certain threshold, which can be fixed or calculated in an adaptive way (as done in the work of Yi et al. [140]), it means that a shot boundary

occurs at that moment of the video. The advantages of using color information for this purpose are the ease of implementation and the descriptive characteristic of both spatial and temporal information [15]. On the other hand, this approach may produce false positives when dealing with videos that contain illumination changes and camera / object motions.

Approaches to SBD include: pixelwise differences [143], color histograms [66], compressed domain techniques [40] and motion vectors [8]. In order to detect as many shot boundaries as possible (regardless of being abrupt or gradual), some other approaches combine different techniques [11, 77, 104].

The disadvantage of the SBD techniques is that they are not applicable to videos that do not have any sort of previous edition (e.g., home videos), since they look for specific transitions between video contents. For this case, specific methods that deal with unstructured video can be applied [78, 87, 136].

2.1.3 Evaluation Metrics

Another challenge in the video summarization field is the definition of standard metrics to evaluate the quality of the results. At the moment, there is no consistent platform to evaluate summaries. Thus, each work has its own evaluation method and, in most cases, it does not compare the results with other existing methods, not only in terms of quality of the produced summaries but also in terms of general performance [124].

One of the most popular metrics to evaluate summaries is by means of precision and recall rates [130], indicating if a given summarization technique generates the summaries in an appropriate way. It is a simple evaluation metric, being often used with a small video database. Another evaluation metric is based on selecting a group of users to judge the quality of the summaries and the degree of satisfaction (usually defined in terms of *informativity* and *enjoyability*). The users then assign a score to each of these attributes and the respective averages over the total number of users are calculated. The advantage of this metric lies in the fact that it is more realistic when compared to precision and recall rates. However, it is more subjective and more difficult to be established, in the sense that it uses people instead of machines.

In order to reduce the subjectivity of evaluation and measure the quality of summaries, Avila et al. [38] developed a metric called CUS (Comparison of User Summaries). In this metric, for a given video, a group of 5 users is asked to manually produce one summary each, selecting the frames that best describe the content of the original video, according to their respective opinions. Then, these summaries are taken as reference (ground-truth) to be compared with the summaries produced automatically by different existing methods. Comparisons are done by taking one frame from the automatic summary and one from a user summary. If a pair of frames is considered similar according to a specific similarity metric (i.e., the similarity between them is above a given threshold), they are removed from the next iteration of the comparison step of CUS.

From these comparisons, two different metrics are defined. The first one is the CUS *accuracy* (CUS_A), which corresponds to the ratio between the number of matching frames from the automatic summaries and the number of frames in the user summary. The

second, is the CUS *error* (CUS_E), which is the ratio between the number of non-matching frames from the automatic summaries and the number of frames in the user summary.

Rather than using CUS_A and CUS_E, an alternative to these metrics is the use of precision and recall rates, which are also a very common method of evaluation. Precision is the ratio of the number of matching frames to the total number of frames in the automatic summary, whereas recall is equivalent to CUS_A. As stated by Almeida et al. [7], since there is a trade-off between precision and recall, the F-measure can be used to assess the quality of the summaries, as defined in Equation 2.1:

$$\text{F-measure} = \frac{2 \times P \times R}{P + R} \tag{2.1}$$

where P and R are the precision and recall rates, respectively.

2.1.4 Databases

In most of the cases, each video summarization method uses its own base of videos to analyze the results, although it does not make any sort of comparison against other methods of the literature, as mentioned in Section 2.1.3. To overcome this issue, Over et al. [99] proposed an automatic framework for evaluation of video summaries with a specific database named TRECVID [4], but it is limited to rush videos. Nevertheless, this database also provides a ground-truth so that the summarization methods can be objectively evaluated, according to the criteria specified by the authors of the framework.

Another well-known database is the Open Video Project (OVP) [3], which consists of a large collection of digital videos of various genres (documentaries, lectures, educational videos, among others) so that researchers can use them to study several problems besides video summarization, such as video retrieval, video annotation and face recognition. Regarding video summarization, the database provides, for each video, a collection of frames that represent the respective contents, which can be useful as a ground-truth for comparing summaries.

To test the CUS metric, Avila et al. [38] used 50 videos from the OVP database. Together, all the video sequences have a total duration of approximately 75 minutes (with each video lasting between 1 and 4 minutes) and 150,000 frames, whose original dimensions are 352 $\times$ 240 pixels. More details about the videos used in this database are listed in Appendix A. From these videos, the authors managed to create a specific database (available at [5]) which contains automatic summaries generated by their method and by a few others [50, 93]. Furthermore, 5 user summaries are also provided as a ground-truth, resulting in 250 user summaries. Thereafter, other authors started to use this database for evaluating their own methods, such as Almeida et al. [7] and Mahmoud et al. [89], thus increasing the number of automatic summaries available for future comparisons.

2.2 Related Work

Zawbaa et al. [142] proposed a soccer video summarization system using SVM (Support Vector Machines [34]), which is one of the most used machine learning techniques. This

system is divided into six steps:

1. *Pre-Processing*: responsible for the segmentation of a given video into several shots, using techniques to detect the dominant color of the field and the shot boundaries.
2. *Shot Processing*: applies two types of classification on the shots generated in the previous step. The first one defines the shot type (based on the visualization), which can be classified as long, medium, close-up and out-of-field (representing the audience); the second one consists of identifying all the plays and interruptions that occur in a match.
3. *Replay Detection*: it is based on the premise that a logo appears during the exhibited replays of a match. Thus, this step detects the replays by means of a logo detection algorithm, using SVM as the classifier.
4. *Scoreboard Detection*: identifies, using SVM, the region of interest that provides information about the match score.
5. *Exciting Event Detection*: once that most of the exciting events of a match happen close to the goal region, such as goals, shoots, fouls and penalties, this step uses an algorithm for detecting the goal posts (using Hough transform), as well as the goal nets (with Gabor filter). Moreover, the audio volume, generated by the commentator or by the present fans, is also taken into account.
6. *Event Detection and Summarization*: make a summary of the input video, containing only the most important events. These events were divided into three categories: goals, attacks and other events (fouls, cards, injuries or off-sides).

The precision rates for the detection of goals, attacks and other events were, respectively, 90.5%, 89% and 97.3%, while the recall rates were 95%, 92.6% and 94%. In general, this approach obtains very satisfactory results. However, it imposes several rules for the replay detection, which can produce undesirable results when using videos of television broadcasts that differently show these replays.

Ekin et al. [45] developed a framework that makes the analysis and the summarization of soccer videos from kinematic and object-based features (such as color, shape, texture, among others). Algorithms for dominant color region detection and shot boundary detection were proposed, also being robust to dominant color variations, considering the fact that the grass color can vary from a stadium to another, besides climatic and illumination factors of a same stadium. Furthermore, new features to classify video shots were proposed, thus providing robustness to variations in kinematic features. Such features rely on dominant color pixel ratio differences and color histogram differences between two frames. Other algorithms that were implemented in this framework include detection of goal, referee and regions close to the penalty areas.

The framework is capable of generating three types of summaries: all slow motion parts, all goals of a game and the slow motion parts classified from object-based features. The first two types only deal with kinematic features of the videos, which are enough for the detection of events (such as goals), leading to a more efficient processing. The third type is used to generate more detailed summaries, improving the accuracy of the results. However, this improvement is obtained by a more expensive computational processing.

At the test execution step, the shot boundary detection algorithm was used to detect two different kinds of shots: cuts (abrupt transitions) and gradual transitions (such as wipes and dissolves). For the cut detection, the framework achieved the average precision and recall rates of 91.7% and 97.3%, respectively, whereas, for the gradual transition detection, these rates were 86.6% and 85.3%. With respect to goal detection, the proposed algorithm runs in real time, achieving precision and recall rates of 45.8% and 90%, respectively, which is considered a satisfactory result for a real time system. Despite of the low precision rate, the recall rate is more important in this case, once the users may be interested in other events besides goals, but they are not tolerant to misdetection of some goals.

Takahashi et al. [122] developed a method for summarization of long sport videos based on metadata, which describe the video contents. From these metadata, each of the plays presented on the videos receives a score based on the degree of significance, the occurrence time and the number of replays of the associated play. The higher the score is related to a scene, the more important it is. After that, the composition of the summary is performed from a selection of a set of the most important scenes from the original video, such that the total time of the scenes from this set does not exceed a limit specified by the user, which can be seen as a combinatorial optimization problem with constraints.

The tests were done with 5 videos of baseball games, with an average time of 3.5 hours each, comparing the summaries generated by the method against the ones produced by the TV broadcasts that provided the original videos. The highest precision and recall rates were 83% and 66%, respectively. The advantage of this method relies on the fact that it can be applied not only on baseball videos, but also on other sports videos that have a similar game structure. However, for each sport, it is necessary to redefine the parameters used to calculate the scores of the play scenes, which are calculated by experimental procedures, in this case.

Bezerra and Lima [16] elaborated a low cost video summarization method by means of visual rhythm [118, 144], from which two feature descriptors based on dominant colors and estimated camera motions were computed. The idea is to detect abrupt pattern changes (related to shot transitions) and texture orientations (related to camera motions). First, the estimation of the field color in the HSV color space was done by identifying the most frequent color in the visual rhythm image. From this color, the detection of the transitions is done from the total number of pixels of the game field. When this number changes abruptly, it means that a shot transition occurs.

For purposes of automatization of this procedure, as well as the classification of the detected shots, the k-means algorithm was used. The shots were separated into two classes: large view scenes and close-up scenes. Then, an estimation of camera motion on the large view scenes was done to detect the direction of the attacks that arise during a match. Such estimative was done based on the visual rhythm image constructed from the central horizontal lines of the frames that compose the shots, once that, to detect this kind of event, it is interesting to observe only the horizontal movements of the camera.

A set of 5 videos of different soccer matches was used in the tests, taking only the first 45 minutes of each match. With the results obtained from the shot classification and camera motion estimative steps, the ball possession times of each team for each game was calculated, with an error of approximately 1% in most of the cases. The average duration

of the summaries was 8.6 minutes, which is considered a reasonable duration time when compared against the 45 minutes of the original videos. In all cases, the results were satisfactory, because all the most important events of each game were detected, which include shots on goal, corner kicks and other relevant events.

Gong and Liu [51] formulated a video summarization system based on Singular Value Decomposition (SVD). Initially, the system generates a feature vector for each video frame, based on 3-D color histograms, using the RGB color space. From the generated vectors, a feature matrix is created for the video. Then, SVD is applied on this matrix, extracting the properties associated to spatial and temporal features of the video from mathematical analysis. Thus, the result achieved with SVD can then be used for the video summarization step.

The evaluation of the summarization stage of the system was done by analyzing three properties: size adjustability and granularity of the summarized video, redundancy reduction and equality of relevance for equal amounts of visual content. The tests were done with a total of 120 minutes of videos, also showing an example in which a 6-minute news report (containing 29 shots) was summarized in an 1-minute video (with 15 shots). For the segmentation of long shots into several groups, the system achieved an accuracy of 86%. In regard to the aggregation of shots with duplicated or similar content, the accuracy was 79%.

Another approach, proposed by Mundur et al. [93], uses the Delaunay Triangulation (DT) [39] to cluster the frames of interest of a video. For each video frame, an HSV-color histogram of 256 bins is constructed. From this histogram, a line-vector of 256 dimensions is created. The composition of all line-vectors, originated from all video frames, forms an $N \times 256$ matrix, where N is the total number of video frames. Next, a Principal Component Analysis [135] is applied to reduce the dimensionality of this matrix, optimizing the total processing time. Later, the DT algorithm is executed on the reduced dimension data, generating the appropriate clusters. The keyframes of each cluster are then identified from their respective centroids.

For the validation of the method, several experiments were conducted with news and documentary videos from the Open Video Project (OVP) database [3], comparing the performance of the clustering algorithm against k-means. The performance evaluation was done by observing three metrics:

- *Significance Factor*: defines the relevance of the cluster contents, taking the number of frames in each cluster as a basis.
- *Overlap Factor*: compares the summaries generated by the method with the ones present at OVP database.
- *Compression Factor*: ratio between the number of keyframes identified by the method and the total number of video frames.

From a total of 50 videos used in the tests, it was noticed that, in 32 of them, the generated summaries contained less keyframes than the summaries provided by OVP, without interfering in the understanding of the original video contents. In addition, in most cases, the overlap factor was close to 100%, which indicates a high likelihood between the summaries generated by the method and the OVP ones.

The main advantage of this method is the absence of pre-defined parameters for the frame clustering step (which is done by most of the clustering methods), once that the Delaunay triangulation of a given set of points located in an n-dimensional space, with $n \in \mathbb{N}$, is unique. On the other hand, the method uses a specific database to evaluate its effectiveness, besides not preserving the temporal order of the generated frames.

Still concerning summarization techniques based on graphs, Ngo et al. [96] developed a method in which a video is represented as an undirected graph. From this graph, a normalized cut algorithm is performed to partition the graph in several shot clusters. Moreover, an attention model is created to determine the best moments of the video based on a structure denominated Motion Vector Field (MVF), used to measure the perceptual quality of the scenes.

Once the clusters and the attention model are obtained from the video, a temporal graph is constructed, being segmented into several subgraphs, in order to obtain a modeling of the scenes that compose the video. The detection of the boundaries between the scenes is done by running Dijkstra's algorithm on the temporal graph, calculating the minimum path between the cluster that contains the first video shot and the one that contains the last shot. Aggregating the result of this step to the attention model, the video summarization is done. The strategy adopted in this step consists of discarding a given fraction of original video frames, observing their contributions to the entropy and the perceptual importance of the output video, which define, together, the quality of the scenes and the clusters, whereas the perceptual information is used to evaluate the quality of the shots and subshots.

The tests were performed with 5 videos, where one is a cartoon, another is an advertising and the other three are home videos. The first two videos contain background audio tracks, unlike the three home videos. For the detection of scene changes, the method obtained precision and recall rates of 87% and 90%, respectively. At the summary validation step, a group of 20 students was chosen to evaluate the quality of the summaries provided by the method in terms of informativity and enjoyability, attributing a score between 0 and 100. For the summaries that correspond to 10% of the original video, the average informativity and enjoyability scores were, respectively, 70.34 and 70.44. For the 25% summaries, the average scores were 82.5 and 80.93.

Benini et al. [14] presented a method for dynamic video summarization using Hidden Markov Models (HMM). The main idea of this work consists of computing motion descriptors to estimate the degree of contribution of each shot of a video in terms of "content informativeness". The composition of the summary is then done by a series of observations of a HMM chain, where each element of this chain represents a story unit, which is a sequence of contiguous and interconnected shots, forming a structure of a semantic scene.

The tests were done using videos of various genres. The general performance of the method was then evaluated the same way as [96], producing summaries that correspond to 10% and 25% of the original video sizes. On average, the informativity and the enjoyability rates for the 10% video summaries were 68% and 72%, respectively, whereas for the 25% summaries, the rates were 81% and 80%. These rates are very significant for a system that deals with lots of video genres. However, the temporal order of the shots is not

preserved in the summaries.

Zhou et al. [147] proposed a dynamic video summarization method by analyzing audio-visual features extracted from videos. The first step of this method consists of the division of a video into a given number of temporal segments. Later, for each video frame, two visual features are extracted: color, obtained from the average values of the color histograms among the three bands of the HSV space; and movement, extracted from SIFT (Scale-Invariant Feature Transform) algorithm [86]). The audio analysis is performed from the extraction of MFCC coefficients (Mel-Frequency Cepstral Coefficients), which are very robust to dintinguish pairs of audio segments.

After this step, the clustering of the segments is done based on the extracted features. In order to fulfill this task, the fuzzy c-means algorithm was used along with an estimative of an optimum number of clusters using EM (Expectation-Maximization) algorithm. Finally, the summarized video is generated by means of sorting and concatenation of the segments. The validation of the proposed method was done with 10 videos of different genres and sizes. On average, the precision and recall rates were 70% and 64.8%, respectively, whereas the informativity and enjoyability rates were 84.6% and 75%, respectively. Such values are considered satisfactory when compared against similar methods.

A different approach from the traditional summarization methods was proposed by Wang et al. [130]. This approach uses a system that works with videos that do not contain any pre-edited structure and with little camera motion (defined by the authors as *aerial videos*), besides using a massive amount of data. At the first part of the summarization, the system extracts the GIST features [49] of each frame. Then, the video is split into several temporal segments using scene classification based on the results obtained from the previous step. After that, the system makes the keyframe extraction of each scene using visual saliency index, which is used to evaluate the value of each frame based on visual attention aspects, achieving results that are closer from those of human perception.

For the temporal segmentation step, the system achieved 70.73% of precision and 85.29% of recall. At the keyframe extraction step, 20 people were selected to evaluate the degree of informativity and enjoyability of the summaries (from 0 to 100). For the summaries corresponding to 5% of the size of the original videos, the obtained results for each factor were, respectively, 91.75 and 77.85. For 10% summaries, the results were 94.25 and 83.05, respectively.

Chasanis et al. [23] elaborated a method for the summarization of rush videos using spectral clustering and sequence alignment. After splitting a given video into shots, a spectral clustering algorithm is executed to extract the keyframes of each shot, which correspond to the cluster medoids (related to the shots) generated by the algorithm, i.e., the frames that present the maximum average similarities with all other frames of a same shot. Next, the keyframes related to color palette and monochromatic frames are discarded. Later, a sequence alignment algorithm is executed to detect the redundancies among the keyframes, followed by the removal of clapboard keyframes by means of the analysis of SIFT features. In the final step, the final summary is produced from the remaining keyframes of the previous step, such that the total time of this summary is lower than a pre-defined duration percentage (associated to the duration of the original video).

A collection of 40 videos from TRECVID [4] was selected for the tests. The duration percentage of the summaries was defined in 2% of the time of the respective original videos. Using the metric described in [99], it was observed that the proposed method produced good results compared to the general average of all approaches that were published at TRECVID 2008. The spectral clustering algorithm used to extract keyframes from the video shots was proven to be very efficient, as well as the one for removing useless frames. However, both clapboard removal and redundancy elimination algorithms had a lower performance than the average.

Furini et al. [50] developed a Web tool that produces both static and dynamic summaries of generic videos on-the-fly. In this tool, users can customize some parameters for the summary, such as the storyboard length and the time they desire to wait to get the storyboard. In the first step of the summarization process, the HSV-color histogram (composed of 256 bins) of each frame is extracted. For dynamic summaries, the tool makes a segmentation of the video considering both audio and video features and then extracts the HSV-color histograms of each segment (which consists of an average histogram of all frames in the segment). Later, a clustering algorithm is executed to group similar frames, selecting a representative for each one. The algorithm used in this work was an improved version of the Farthest Point-First (FPF) algorithm [53], which was adapted to the video summarization context.

The tests were conducted with short videos (including ones from OVP [3]) and long videos of several genres. To evaluate the qualities of the summaries, the Mean Opinion Score (MOS) metric was used. First, a group of 20 users was selected to evaluate the quality of each summary, varying from 1 (bad quality) to 5 (good quality). Then, the MOS is computed from the average of the scores of all users. The results were compared against [93], OVP and the standard k-means approach, having the best MOS for most video genres. In addition to the quality measure, the general performance was also evaluated. Compared to the k-means approach, it achieved an average speedup of 5 times for the dynamic summarization and average speedups of 55 times for long videos and 25 times for short videos regarding the static summarization. The advantage of this approach relies on its fast clustering algorithm, which makes it suitable for an on-the-fly usage. Furthermore, the achieved speedups do not compromise the quality of the summaries. On the other hand, it still generates summaries with worse quality than a random approach to some specific genres, such as news and talk-shows.

Avila et al. [38] proposed a methodology for static video summarization called VSUMM (Video SUMMarization). After sampling the video frames at one frame per second (considering videos of 30 frames per second), the method extracts the features from the remaining frames based on the 16-bin color histogram of Hue values obtained from HSV color space of each frame. The frame clustering process is done by k-means, where k is estimated from a shot boundary detection algorithm described in [54]. Moreover, it presents a new evaluation method, called Comparison of User Summaries (CUS, which was detailed in Section 2.1.3), where a number of users manually produce the summaries, which are then taken as a ground-truth for comparison against automatic methods (DT [93] and STIMO [50] were chosen for this purpose).

The evaluation of the summaries was done using 50 videos of various genres from OVP. Moreover, two versions of VSUMM (labelled as VSUMM_1 and VSUMM_2) were tested, where one is slightly different from another concerning the clustering step. In VSUMM_1, one keyframe is obtained per cluster, whereas in VSUMM_2 one keyframe is obtained per keycluster (a cluster whose size is larger than half the average cluster size). Using the CUS metric, it was observed that VSUMM_1 achieved the highest accuracy rates and VSUMM_2 the lowest error rates. Also, VSUMM_1 presented a higher error rate than DT approach and VSUMM_2 obtained a lower accuracy rate than STIMO and VSUMM_1. Therefore, VSUMM_1 provided the best results.

Almeida et al. [7] developed a video summarization approach, named VISON, which works directly on the compressed domain and allows user interaction to control the quality of the summaries. This approach extracts features from videos by computing reduced versions of the frames, defined as DC images (derived from the DC terms of all 8×8 pixel blocks of the respective frames), thus saving some computational time. It also saves space for the images by calculating 256-dimensional feature vectors from each DC image, which is done from the extraction of HSV color histograms. At the next step, the content selection is done from the extracted histograms, grouping frames that have similar contents using the Zero-mean Normalized Cross Correlation (ZNCC) metric to measure the distances between all pairs of frames. This metric was chosen due to its robustness to changes in photometric parameters, such as brightness and contrast. Finally, the approach performs a noise filtering step to avoid the inclusion of redundant or meaningless frames in a summary.

Similar to [38], the CUS metric was used to evaluate the summaries generated by the method, comparing the results not only against VSUMM, but also with DT and STIMO, besides using the same database. Furthermore, to measure the quality of summaries, the F-measure metric was adopted, rather than using precision and recall rates apart, due to the existing trade-off between these two measures. For all the 50 videos from OVP, VISON achieved the highest F-measure values among the aforementioned methods. Comparing VISON summaries against the ones made by users (the same used by [38] as a ground-truth), it was noticed that the summaries were very close. The advantage of this approach lies on its computational efficiency, at the same time it generates high quality summaries. On the other hand, it achieves low F-measure values for specific genres of videos, such as sports and commercials.

Another approach using the same ground-truth and evaluation metric for the tests was proposed by Mahmoud et al. [89]. In this work, frames are described using both color and texture features by using HSV-color histograms and the Discrete Haar Wavelet Transform (DWT) [67], respectively. The Bhattacharyya distance [68] was used to measure the similarity between video frames, due to its consistency and the fact that it is not affected by the distribution of the data along the histogram. The keyframe extraction process is done by means of a variation of DBSCAN [47], a density-based clustering algorithm that, unlike K-means, does not require the number of clusters to be known *a priori*. Once the frames of a given video are clustered, the middle core frames (in ordered sequence) of each cluster are chosen as the keyframes for the final summary.

When comparing the summaries against different approaches with the CUS metric, the authors used their own image descriptor along with the Bhattacharyya distance to identify the similarities. In addition, they claim that a more perceptual assessment of the quality of the summaries is assured because of the use of color and texture features, instead of using just color, as done in the two previous approaches. Using the same F-measure metric from [7], the approach achieved a mean value of 0.77, thus performing better than the other tested methods (VSUMM, DT and STIMO).

With respect to recent approaches for video summarization, Sharma and Sathish [114] developed a method that parallelizes the keyframe extraction process in GPUs (Graphics Processing Units). Such process is done by applying DWT on each video frame, thus extracting four different sub-bands (LL, LH, HL and HH) and comparing these bands with a threshold in order to classify that frame as a keyframe or not. The parallelization of the method was done using the CUDA platform (Compute Unified Device Architecture) [97]. Tests were conducted with HD videos (1920 $\times$ 1080 pixels for each frame), where the GPU implementation achieved an average speedup of 60% over the CPU version.

Kavitha and Rani [69] developed a priority fusion method for video summarization that combines static and dynamic features, hence becoming suitable for both slow motion and fast moving videos. After running a shot boundary detection algorithm, the static features are created by dividing each video frame in 8×8 blocks and converting them to the LMS color space [110]. Then, a static attention value and the saliency map of each frame are calculated, giving a set of salient keyframes for each detected shot. The dynamic features are obtained by running DWT on each shot, giving a DWT attention value which is then used to retrieve the most interesting events from the video. Finally, the keyframes selected in both stages are combined to generate the video summary. The method was tested with a set of 5 videos from the OVP database using the F-measure metric to assess the quality of the produced summaries. From these videos, the method achieved an average F-measure of 0.8 and an accuracy of 99% regarding the detection of the most important frames according to the ground-truth.

Chu et al. [28] presented a different perspective for summarization of videos that exploits the fact that important video concepts tend to appear repeatedly across different videos. Such method is called *video co-summarization*. From an input video of a specific query topic, the method determines the importance of each shot by fetching visual co-occurring shots in additional retrieved videos of that topic. In other words, it looks for shots that co-occur most frequently across different videos of a same topic. However, since the common patterns are scattered along with several irrelevant shots from all videos, an effective algorithm that only retrieves shots from the desired topic must be developed. For this purpose, the authors developed a Maximal Biclique Finding (MBF) algorithm, which deals with the sparsity of co-occurring shots by discarding the ones that only appear in a single video.

The tests were conducted on groups of videos of several topics from YouTube dataset [6], where each group contained between 3 and 7 videos from a same topic. Then, a group of 20 subjects was asked to evaluate the summaries generated from the proposed method, along with the ones produced by three other approaches. For each topic, each subject had to label the summaries of each method as good (+1), neutral (0) or

bad (-1), which defines the relevance between the summary content and the respective topic. Furthermore, an objective evaluation was performed, in terms of the F-measure score, by using a specific database that contains 10 different human actions (used as the ground-truth). In both cases, MBF produced better results than the other methods.

Chapter 3

Video Summarization by Spectral Clustering

This chapter discusses the approach proposed in [31], which performs video summarization using a special kind of clustering technique, called spectral clustering. Firstly, general concepts about this technique are described, along with different image descriptors used to represent the video frames. Then, the applied methodology is thoroughly analyzed, describing the steps used in the whole video summarization process.

3.1 Spectral Clustering

Spectral clustering [88, 95, 127] is a method derived from the spectral graph theory [29] and can be defined as a method that clusters data points using eigenvectors of matrices calculated from a given dataset. It is an important research object in many fields such as pattern recognition, machine learning, and signal processing. Concerning the video summarization context, it can be used in several tasks, including keyframe extraction [23], shot boundary detection [37] and important events detection [36]. Furthermore, it can be easily implemented using platforms that deal efficiently with linear algebra operations (e.g. MATLAB [1]) and usually outperforms traditional clustering algorithms, including K-means [112]).

A general spectral clustering algorithm can be implemented as follows: given a set of n points (representing some sort of data) located at an l-dimensional space to be divided into k distinct subsets, where n, l and k are positive integers, an affinity matrix $A_{n\times n}$ is constructed such that each element $A(i,j)$ corresponds to a similarity measure $s_{ij} \geq 0$ that represents the likelihood degree between a pair of points i and j of the set, with $i \neq j$ and $A(i,i) = 0$. Thus, the higher the value of $A(i,j)$, the higher is the similarity between the points i and j and vice-versa. Without any loss of generality, let the similarity measure range between 0 and 1, where the latter corresponds to the case when points i and j store exactly the same information.

Then, the diagonal matrix $D_{n\times n}$ and the Laplacian matrix L are defined according to

Equations 3.1 and 3.2, respectively:

$$D(i,i) = \sum_{j=1}^{n} A(i,j) \tag{3.1}$$

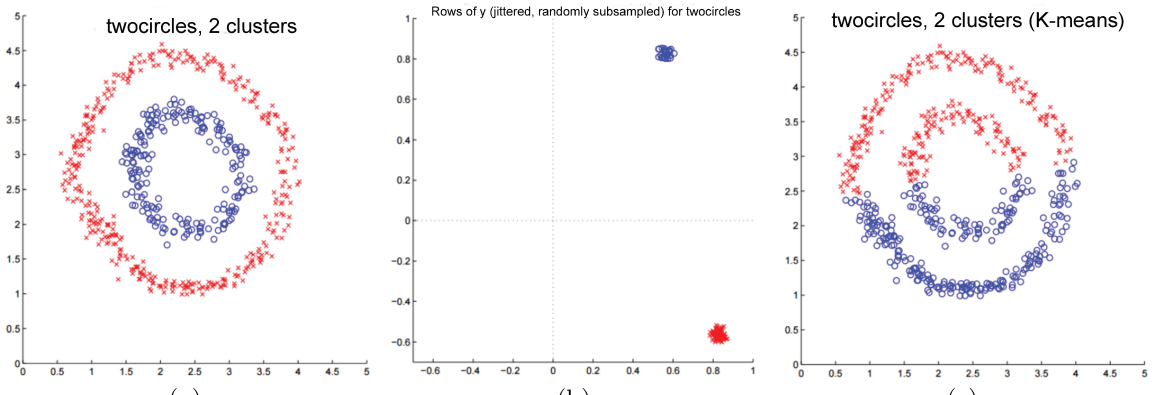
$$L = I - (D^{-1/2} A D^{-1/2}) \tag{3.2}$$

where $I_{n \times n}$ is the identity matrix. In the next step, the k largest eigenvectors of L are calculated, forming the matrix $X_{n \times k} = [x_1 \; x_2 \; ... \; x_k]$ by stacking these eigenvectors in k columns. After that, the matrix $Y_{n \times k}$ is created from X by normalizing the rows of X such that each one has unitary length, as specified in Equation 3.3.

$$Y(i,j) = \frac{X(i,j)}{\sqrt{\sum_{j=1}^{n} (X(i,j))^2}} \tag{3.3}$$

Finally, treating each row of Y as a point in $\mathbb{R}^k$, the rows of Y are separated into k groups by the K-means algorithm (or any other clustering algorithm, such as the ones described in [46]) assigning the point i of the initial set to group j if, and only if, the row i of matrix Y is assigned to cluster j.

It is important to observe the key difference between applying the K-means algorithm on the normalized matrix of eigenvectors and applying K-means directly on the original dataset. Spectral clustering is more suitable for finding clusters that have non-convex boundaries, as can be seen in Figure 3.1. This works because the points located in $\mathbb{R}^2$ do not form convex regions, which produces an unsatisfactory result when running K-means directly on this domain. On the other hand, when the points are mapped in the $\mathbb{R}^k$, they form tight clusters, thus being way easier to identify the clusters correctly.



(a) (b) (c)

Figure 3.1: Comparison of spectral clustering (a) against K-means (c). Figure (b) shows the points of (a) mapped in $\mathbb{R}^k$. Images extracted from [117].

The spectral clustering problem can also be formulated from the point of view of graphs. In this case, a similarity graph G (represented by an adjacency matrix) is constructed, whose vertices correspond to the points of the initial set and its edges define the

similarity degree between pairs of vertices, based on the affinity matrix A. Two vertices i and j are connected if $A(i, j)$ (equivalent to the weight of the edge that connects them) is positive or larger than a certain threshold. Therefore, to split the vertices of G into k groups, it is desirable to find a partition of G such that edges between vertices of different groups have high weights (low similarity), at the same time that edges between vertices from a same group have low weights (high similarity). However, finding such partition is an NP-hard problem [19, 29, 128]. Thus, spectral clustering plays a fundamental role in this task by solving relaxed versions of this problem. The most used techniques for this procedure are Normalized Cuts [117] and Ratio Cut [56].

Defining both the similarity measure and the number of clusters in which the dataset is split is not a trivial task, once that they are subject to the application domain from where the dataset comes. First of all, it must be assured that the data considered as "very similar" by the chosen similarity measure has a very close relationship in the application domain as well [88]. Moreover, in most cases, there is not a "correct" number of groups. In this situation, it is common to use strategies that find this number in an automatic way [90, 113].

A drawback of spectral clustering algorithms is that the computed matrices are very large, demanding a large storage space, especially when working with digital videos, composed of a considerable number of frames. In order to guarantee the efficiency on the implementation of these algorithms, the Laplacian matrix L must be sparse, simplifying the task of calculating the k largest eigenvectors and avoiding the computation of the similarity measures between every single pair of points. For this procedure, graphs such as ϵ-neighborhood and k-nearest neighbors are used.

3.2 Local Feature Extraction

One of the most recurrent challenges in computer vision applications, such as object recognition and matching, 3D scene reconstruction and motion tracking, is the representation of images by finding points of interest (*keypoints*), from which many relevant pieces of information, called *features*, can be extracted. According to Bay et al. [13], keypoint detection algorithms try to look for salient image regions (e.g. corners, blobs, T-junctions) in such a way they must be repeatably found, regardless of illumination and viewpoint changes. At the same time, algorithms that extract descriptors from those keypoints capture the most important information, such that it must be distinctive and robust to noise, as well as to detection errors and geometric and photometric changes. When matching two different images, the respective descriptors (represented by vectors) are compared by a specific distance function, generally the Euclidean or the Mahalanobis distance. The performance of this task depends strictly on the number of detected keypoints and the number of dimensions of the chosen descriptor (i.e., size of the feature vectors).

Two of the most popular algorithms for both keypoint detection and feature extraction are SIFT [86] and SURF (Speeded-Up Robust Features) [13]. Both of them transform a given image into a large collection of local feature vectors that are invariant to several image transformations, which include scale, translation, rotation and affine or 3D

projections. Regarding SIFT, keypoints are firstly detected in a scale-space by looking for maximum and minimum locations of a Difference-of-Gaussian function in order to identify potential candidates that are invariant to both scale and local transformations. Then, the rotation invariance is assured by attributing one or many orientations to each detected keypoint based on local image gradients. Finally, for each keypoint, a descriptor is generated by using a set of orientation histograms, created by sampling each keypoint over a 4×4 neighborhood grid where each cell contains 8 bins, leading to a total of 128 dimensions for each descriptor. However, such descriptor size can compromise the efficiency of image comparison processes, which was a motivation to develop faster solutions for this algorithm, such as PCA-SIFT [70], which drastically reduces the size of the feature vectors by selecting the "most relevant" dimensions.

On the other hand, SURF uses different and faster approaches for both keypoint detection and feature extraction. The former procedure is done by using Hessian matrices [62], whose determinants measure local changes around each detected keypoint at the same time they define the scales of these keypoints. In order to guarantee the rotation invariance, Haar-Wavelet responses in both horizontal and vertical directions around each keypoint are computed. Then, each orientation is obtained by calculating the sum of all responses within a sliding orientation window. For the latter procedure, a square region centered around each keypoint is constructed, which is oriented along the orientation calculated in the previous step. Later, this region is split into 4×4 smaller subregions. For each subregion, 4 different sums of wavelet responses are calculated: one for each 2D-direction ($\sum dx$ and $\sum dy$) and their absolute values ($\sum |dx|$ and $\sum |dy|$). This leads to a feature vector of 64 dimensions, thus becoming much faster than SIFT for computation and eventual matchings.

Another approach for the extraction of local features is the use of binary descriptors [60], which aim for computation efficiency and provide a compact representation, thus becoming suitable for platforms that demand little hardware requirements, such as mobile applications. For this approach, descriptors are computed by a set of direct pixel-level comparisons of pairwise intensities. The results of all comparisons are represented by respective bits, which comprise a string of bits of fixed size that represents the feature vector of a point-of-interest in an image. Furthermore, rather than using Euclidean or Mahalanobis distance for matching purpose, the Hamming distance, which compares strings of bits using logical operations, is used, becoming much faster for this task than SIFT and SURF. Some examples of binary image descriptors include: BRIEF (Binary Robust Independent Elementary Features) [20], ORB (Oriented FAST and Rotated BRIEF) [111] and BRISK (Binary Robust Invariant Scalable Keypoints) [79].

3.3 Proposed Methodology

Figure 3.2 shows a general flowchart of the proposed method of video summarization with spectral clustering. Given a digital video, the first step of the method consists of its sampling in a smaller number of frames, in order to increase performance. For this work, a sampling of 5 frames per second as performed. At the feature extraction step,

image descriptors for each sampled frame are calculated by means of one of the methods described in Section 3.2.

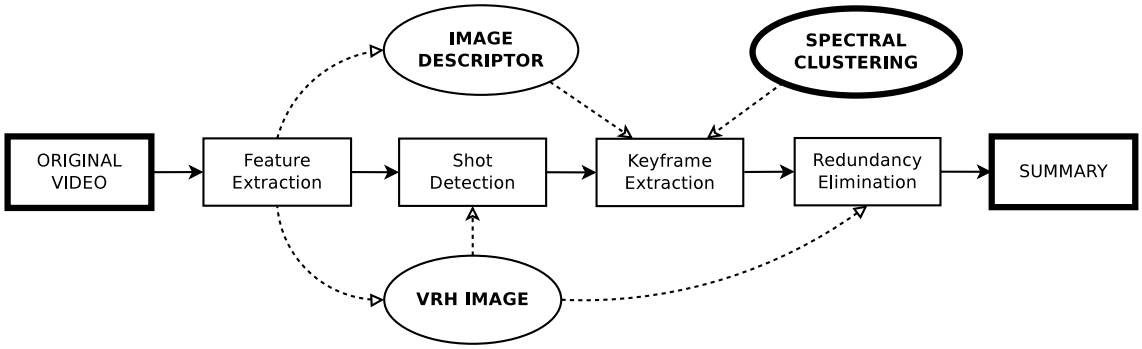


Figure 3.2: Flowchart of the main stages of the method proposed in [31].

Furthermore, a visual rhythm by histogram (VRH) [54] image is computed as shown in Figure 3.3. From each of the N sampled frames, a grayscale histogram of L bins is calculated, whose values are normalized to a scale between 0 and 255. Then, these histograms are consecutively disposed in columns in order to form a new image VRH_I (also in grayscale), where the histogram of the first frame is the first column of VRH_I, the histogram of the second frame is the second column, and so on, thus forming an image of size $N \times L$. Looking at the example for the video *The Great Web of Water, Segment 02*, the transitions between the shots of the video can be seen at changes in the patterns of the image, where the cuts, for instance, are represented by "vertical lines". Therefore, the task of identifying the shot boundaries can be done by pixelwise differences of columns of the VRH image, rather than comparing the original images.

Next, at the shot detection step, the estimation of the number of video shots is performed, based on a shot boundary detection algorithm. From VRH_I, the shot boundaries are detected by using the local adaptive threshold technique described in [116]. Rather than comparing the sampled frames, the columns of VRH_I are used in this process. Let k be the total number of video shots. Starting with $k = 1$, every time a shot boundary is detected, k is incremented by 1.

Then, the estimated value of k will be used at the keyframe extraction step. In this step, a spectral clustering algorithm is executed. First, an affinity matrix A is constructed, as defined in Section 3.1, where the element $A(i, j)$ corresponds to the distance between the image descriptors of frames i and j. It is important to notice that the term "distance" is being used instead of "similarity". When constructing the affinity matrix, the only difference is that frames i and j are exactly the same if, and only if, $A(i, j) = 0$. Before computing the distance between frames i and j, all the matches between keypoints of these frames are analyzed by computing the Euclidean distances between all pairs of keypoints. However, a substantial part of these matches are "bad matches", meaning that they do not necessarily represent the same structural information in both images. Hence, only the "good matches" are analyzed in this process. Taking the minimum computed distance between all matches as a base value, if the distance between two matching keypoints is equal to or less than three times the minimum distance, that matching is regarded as

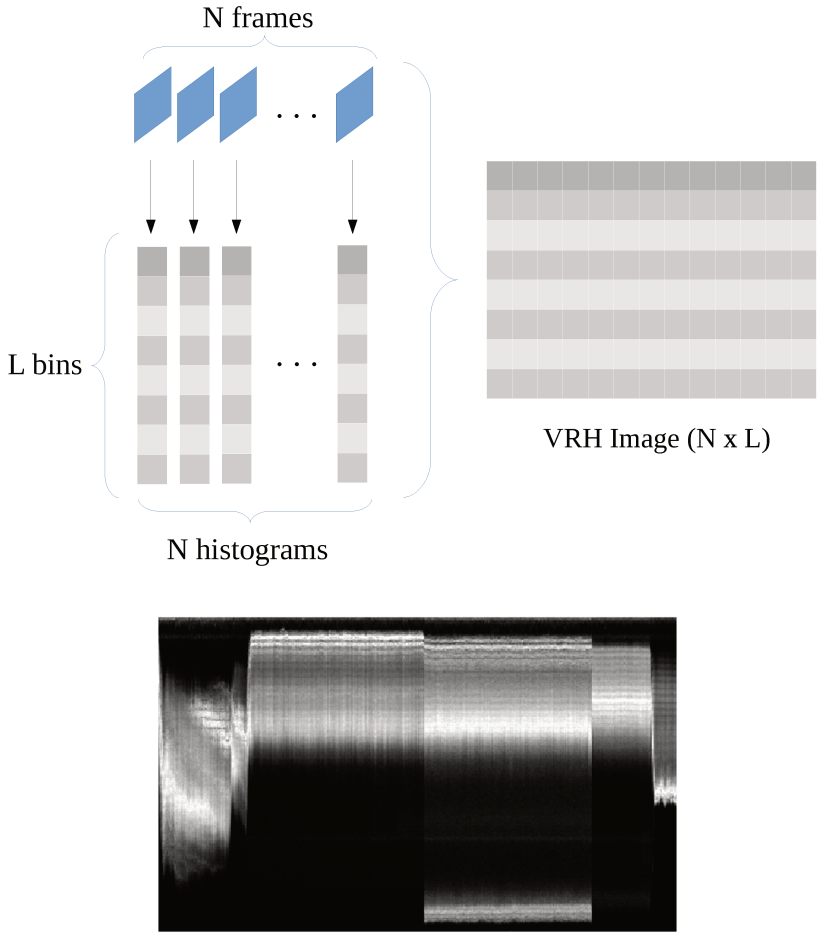


Figure 3.3: General scheme of how a VRH image is computed, along with an example using the video *The Great Web of Water, Segment 02*.

"good". Finally, the distance between frames i and j is defined as the average distance of all "good matches", which is then assigned to $A(i, j)$. If there is no match between i and j, $A(i, j) = 1$.

Once A is constructed, the following steps of the spectral clustering algorithm are executed. After the calculation of the normalized eigenvectors, the K-means algorithm is executed to cluster the frames according to the shots which they are related to, where the number of clusters corresponds to k. Later, the keyframes of each cluster are extracted based on the centroids calculated by K-means (one keyframe per cluster). The chosen keyframes correspond to the ones that are closest to their respective cluster centroids.

At the next step of the proposed method, a post-processing is then performed to eliminate redundant frames. This step is performed by computing the sums of pairwise pixel distances between the columns of the VRH image (generated in the feature extraction stage) related to two consecutive keyframes. After that, these values are compared to a distance threshold T_d. If the distance between keyframe i and keyframe $i + 1$, where $1 \leq i \leq k - 1$, is less than T_d, the keyframe i will be considered as redundant and, therefore, will not be included in the final summary. The threshold value was empirically

defined as $T_d = (\mu_d + \sigma_d)/4$, where μ_d and σ_d are the mean and the standard deviation of all distances, respectively. This approach performs well with most of the generated redundant frames from the videos used in the tests, but it may fail at detecting redundant frames with high luminosity differences (brightness and contrast), since their columns in the VRH image are very distant from each other. From the remaining keyframes, the final summary is then created.

The advantage of this method is that every stage is executed in an unsupervised fashion, such that the number of shots does not need to be known *a priori*. However, the whole summarization process is still expensive, because of the spectral clustering, even though it leads to more accurate results than standard clustering approaches.

3.4 Experimental Results

The tests were conducted using an AMD Phenom II X6 3.2 GHz processor and 4 GB of memory. The methodology described in Section 3.3 was implemented with OpenCV platform [2]. Also, the OVP database (see Section 2.1.4) was used for the tests.

Regarding the image descriptors, only three of them were used in this work: SIFT, SURF and ORB. According to Canclini et al. [21], among these three descriptors, ORB provides the fastest computation time for both keypoint detection and feature extraction, followed by SURF and SIFT. However, after the execution of the implementation of the proposed method for each descriptor and using all videos, it was observed that SIFT provided the fastest execution time, with a total execution time of 1.10 hours, followed by ORB (4.04 hours) and SURF (7.59 hours). The reason for that is because the performance bottleneck of the proposed method lies on the number of detected keypoints for each frame, where the higher this number is, the higher is the amount of pairwise comparisons between keypoints of different frames.

The evaluation of the quality of the summaries in this work was performed in a subjective way, based on a strict comparison of the amount of informative content that was included or not in the final summaries, taking the summaries from OVP as the ground-truth. Two videos were taken as examples: *The Great Web of Water, Segment 02*, which has 5 shots, and *Hurricane Force - A Coastal Perspective, Segment 03*, with 12 shots. Figures 3.4 and 3.5 show the respective results, along with the summaries generated by different approaches, as well as the one provided by the OVP database, which was defined as the ground-truth. For the first video, it can be seen that the proposed method generated summaries with 6 keyframes, one more than the number of shots, which means the shot boundary detection process performed very well for this video. Also, the redundant frames (represented as grayscale images) were properly detected and eliminated for the final summary, once that the respective contents of the detected redundant frames are similar to their consecutive frames, leaving only the colored ones. With respect to the quality of the summaries, the SURF-based summary was the only that included the contents of all shots, being the closest to the OVP summary. Furthermore, the SIFT-based summary included two keyframes of a same shot (1st and 2nd frames), and the ORB summary was the one that generated more redundant frames (2nd and 4th frames) than

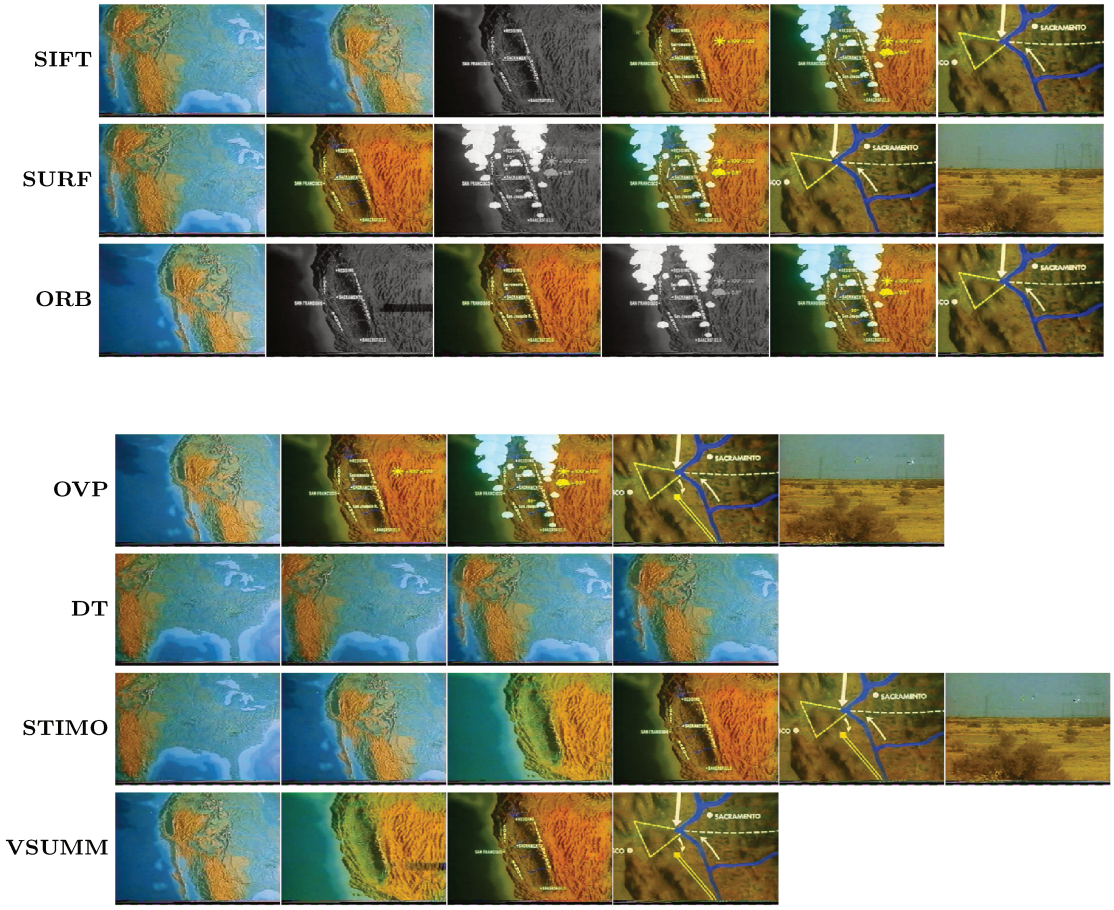


Figure 3.4: Summarization results for *The Great Web of Water, Segment 02*) video. For each descriptor, redundant frames are represented as grayscale images.

the other descriptors. Comparing to other approaches, SURF performed slightly better than both STIMO and VSUMM, which produced the best summaries among the other approaches. This happens because STIMO included more than one frame of a shot, even though it included at least one frame of every shot, and VSUMM missed the last shot.

For the second video, all the summaries of each descriptor contain 11 keyframes, one less than the number of video shots. In the redundancy elimination process, it can be noticed that three frames were discarded in the ORB summary (1st, 6th and 7th frames), whereas SIFT summary had two discarded frames (1st and 8th frames) and only one for the SURF summary (7th frame). However, all the eliminated frames (except for the 7th one of the ORB summary) have little more information than the remaining consecutive frames of the respective final summaries. Concerning the summary content, SURF selected most of the different shots not only among the descriptors but also the other approaches as well. On the other hand, comparing the summaries of the proposed method to the OVP summary, none of them was able to select a frame from the first shot, as occurred both in DT and VSUMM summaries.

In general, it is hard to evaluate how the misdetection of a shot (i.e., when a frame of a shot is not included in the final summary) affects the comprehension of the central message transmitted by a video. For that, a more subjective evaluation must be made, once it requires a deeper content analysis and a general consensus about the degree of

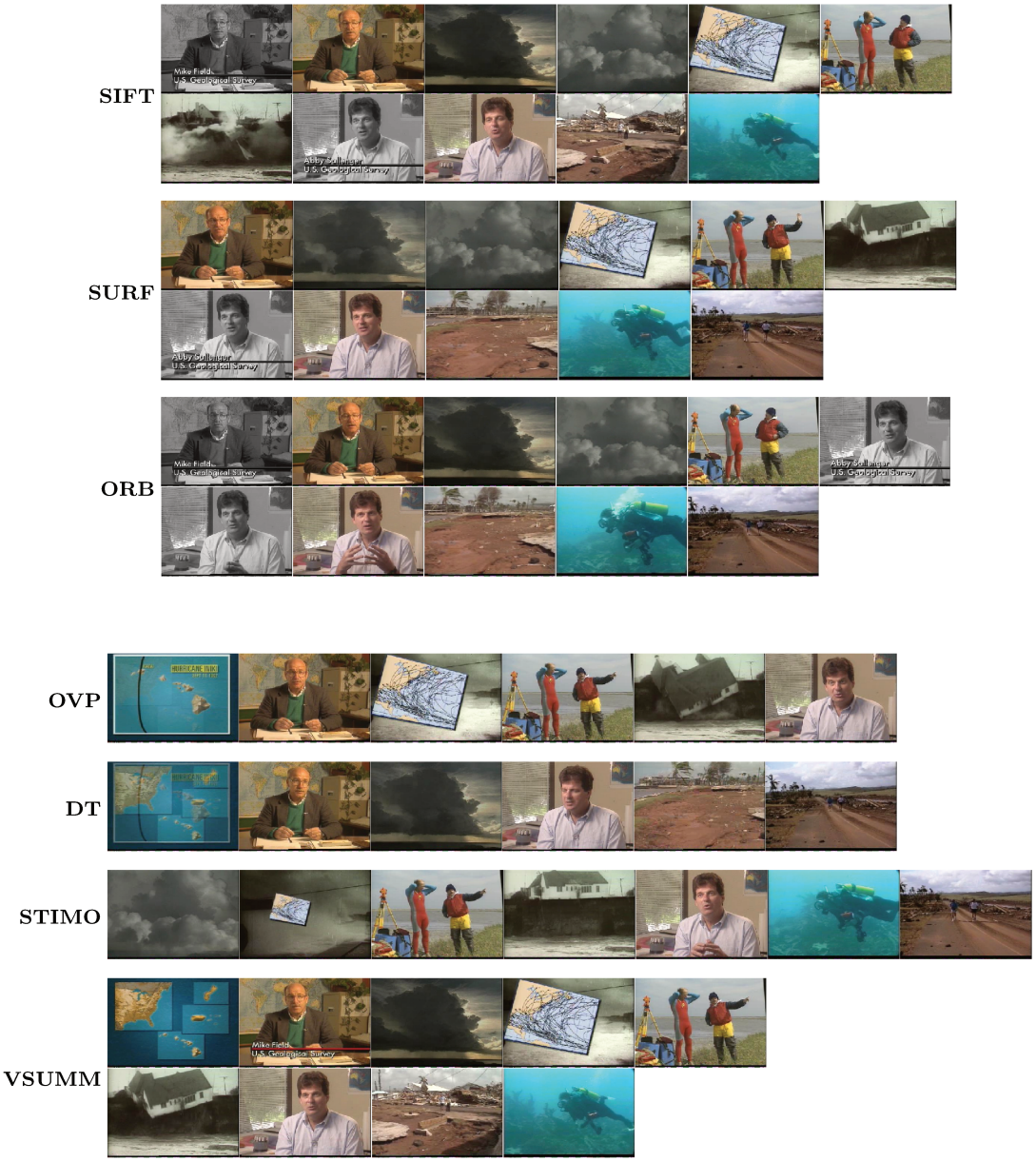


Figure 3.5: Summarization results for *Hurricane Force - A Coastal Perspective, Segment 03*. For each descriptor, redundant frames are represented as grayscale images.

relevance of each shot. In other words, even though the summaries produced by each descriptor have more different shots than the ones of other approaches (including the OVP), all of them may have the same relevance in particular situations.

3.5 Discussion

Despite the slowest processing time, the summaries produced by SURF were the best among the tested descriptors, once they detected most of the different shots and generated less redundant frames than SIFT and ORB. Comparing SURF to other approaches, the results were very close in most cases, although SURF produced more complete summaries in terms of informability.

Even though the generated results were satisfactory, there is still a need for a well-defined similarity metric for image comparison, as well as a more consistent method of evaluation, once the analysis was strictly subjective. These will be the general guidelines for the following method.

Chapter 4

Video Summarization by Image Quality Assessment

This chapter describes the method proposed in [32], named VSQUAL, which uses image quality assessment metrics as a similarity measure for pairwise comparison of video frames. Concepts about those metrics are detailed, as well as the definition of the most used ones. Applications in video summarization are also discussed, comparing the differences in the pipeline stages against the method described in Chapter 3.

4.1 Image Quality Assessment (IQA)

When dealing with image quality evaluation, in order to achieve the highest possible confidentiability, an ideal application would use human beings to give their opinions about the quality of images, looking for distortions which are often caused by problems in image acquisition, processing, storage, compression or transmission. However, such procedure is very time-consuming and expensive, thus being not suitable for real-time applications. Moreover, this kind of evaluation is subjective, meaning that each person perceives image quality by his/her own. Therefore, to overcome the aforementioned issues, there is a need for developing objective metrics that measure the quality of images in a faster and automatic way, resulting in a quality measure that consistently represents the way that humans perceive those images.

In order to do this, several objective IQA metrics [26, 131, 132, 145] have been developed in the last decades, being useful in applications such as image and video coding [139], digital watermarking [73] and image synthesis [129]. Their main characteristic is the exploitation of physiological and psychophysical characteristics of the human visual system (HVS), at the same time they take into account the structural information of images. A complete survey describing those metrics can be seen at [22].

According to [132], objective IQA metrics can be classified into three different categories, depending on the availability of an original image with no distortions, which is the main reference used in comparisons to distorted images. Those categories include:

- **Full-Reference (FR)**: comprising the vast majority of IQA metrics, FR-based techniques take both a reference image and a distorted image as an input, giving

an estimation of the quality of the latter related to the former. The simplest way to achieve this is by means of pixelwise differences, which is done by methods such as MSE (Mean Square Error) and PSNR (Peak Signal-to-Noise Ratio) [63]. However, the results obtained by those metrics show a poor correlation with human perception when compared to other FR methods, especially the ones that take into account the structural similarity of images, such as UQI (Universal Image Quality Index) [131] and SSIM (Structural Similarity Index) [132]. Some extensions of SSIM have been proposed as well, including, but not limited to, a multiscale version called MS-SSIM [134], the perceptual version PSSIM [108] and the feature-based version FSIM [145] (which will be analyzed in the next section). Another well-known FR method is VIF (Visual Information Fidelity) [115], which operates on statistical information of natural scenes.

- **Reduced-Reference (RR)**: techniques of this type provide partial information (represented by a set of features) about the reference image. Nevertheless, this information is still useful to predict image quality along with the distorted image. According to Li and Wang [82], despite the choice of the features for the reference image is flexible, it must satisfy 3 conditions: 1) provide a good summary of the reference image; 2) be sensitive to several image distortions; and 3) be relevant to the visual perception of image quality. Examples of RR methods include an adapted version of SSIM [109] and the use of DWT (Discrete Wavelet Transform) coefficients for the set of features of the reference images [94, 121, 133]

- **No-Reference (NR) or Blind-Reference**: as the name suggests, NR methods do not need reference images in order to measure image quality. Instead, they look for specific distortions in images, such as blurring [42], sharpness [146], blocking [25], ringing [85] or other types of image noise.

A drawback of objective IQA metrics is their high sensitivity to geometric changes, such as translation, scaling, rotation, and so on. Unless these changes are not too drastic, it can cause a huge impact on the measured quality between the reference and the distorted image, due to massive changes in the pairwise pixel intensities.

For this work, the FSIM metric was used as the similarity measure. FSIM is a FR-IQA method proposed by Zhang et al. [145] which was designed from the principle that the HVS interprets a scene by analyzing the information contained in salient low-level features, such as edges and zero-crossings [92]. The image quality is measured from two complementary features:

- **Phase Congruence (PC)**: regarded as the primary feature of FSIM, it is a dimensionless measure that estimates the significance of a local structure in the image. Rather than detecting sudden changes in pixel intensities, the computation of this feature is done by looking for points in an image where its Fourier components have a maximum phase. In turn, these points represent features that are discernable to the HVS, hence providing a plausible model of visual perception, which is also invariant to contrast.

- **Gradient Magnitude (GM)**: even though PC is contrast invariant, image local contrast affects the perception of image quality. Thus, GM is computed as the secondary feature of FSIM to encode contrast information. It is calculated from the partial derivatives along horizontal and vertical directions using three different operators: Sobel, Prewitt and Scharr [52, 65].

Unlike the majority of IQA metrics, FSIM can be easily extended to work with color images, leading to a new measure called FSIM_C. Since color information is a fundamental part of image and scene understanding, it is expected that FSIM performs better than other approaches. In order to achieve that, images are converted to YIQ color space, where luminance (Y) and chrominance (I and Q) components can be separated.

4.2 Proposed Methodology

Figure 4.1 shows a flowchart of the VSQUAL method. Like the previous approach described in Section 3.3, the original video is sampled in a smaller amount of frames (5 frames per second). From these frames, the similarity matrix S is computed, where, for each pair of frames i and j, $S(i,j) = \text{FSIM}_C(i,j)$. The values of each position $S(i,j)$ range from 0 to 1, where 1 indicates that images i and j are exactly the same, and 0, that they are completely distant from each other.

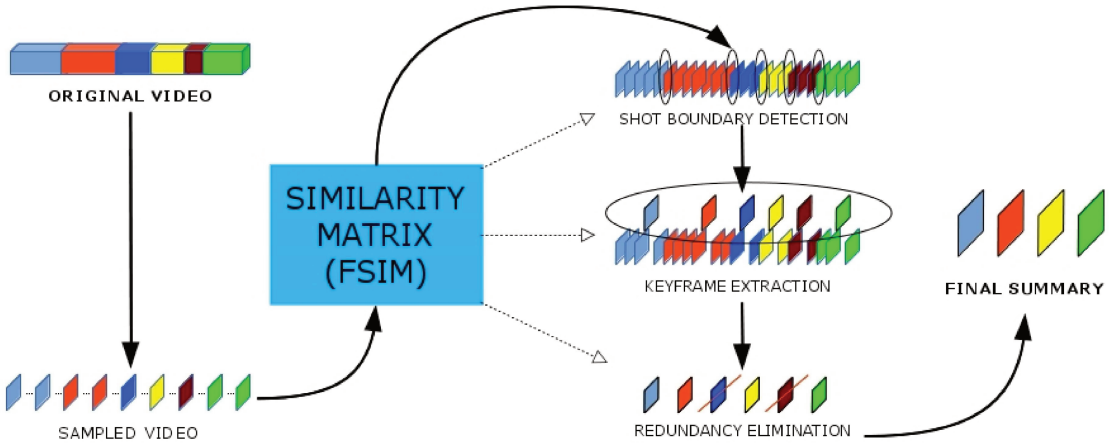


Figure 4.1: Flowchart of the main stages of VSQUAL, proposed in [32].

After that, at the shot detection stage, the number of video shots is estimated by analyzing the FSIM_C's between consecutive frames, which correspond to the set of all $S(i,i+1)$ elements such that $i = 1, 2, ..., N-1$, where N is the total number of sampled frames. Figure 4.2 shows a plot of how the values of these elements vary. When the values stay high at some interval, it is more likely that all frames from that interval belong to the same shot. This happens because consecutive frames of a shot generally have similar contents and, therefore, high FSIM_C values. In addition, whenever there is a significant decreasing in the FSIM_C values, it means that a shot transition may be detected, since

the contents of a different shot are about to appear at a video. Then, a new interval of high values appears, which represents the contents of a new shot.

Hence, the shot boundary detection problem can be reduced to the problem of finding points of local minima in a sequence of FSIM_C between consecutive frames. However, this must be done such a way that different kinds of shot boundaries can be detected, since the mere analysis of the absolute values of the plot is more suitable for detecting only the abrupt transitions. So, a sliding window of size m is used in order to look for large variations in the sequence and detect multiple kinds of shot boundaries. Starting from $j = 1$, all points from j to $j + m - 1$ are inserted into the window. Then, if the middle value is the lowest among all points and the window amplitude, i.e. the difference between the maximum and the minimum values of the window, is greater than a threshold T_B, the middle point will be regarded as a shot boundary. The threshold is necessary so that it does not consider every single minimum point as a boundary. In this process, the values for m and T_B were empirically defined as 9 and 0.1, respectively.

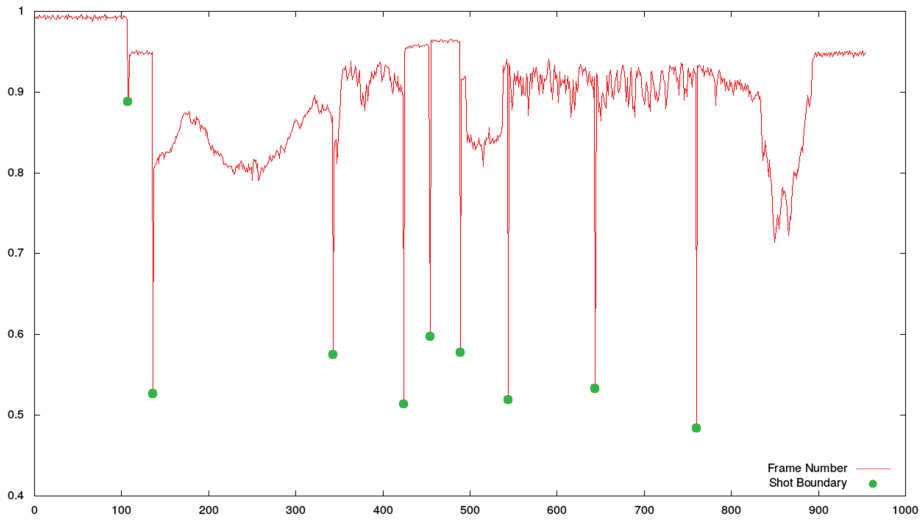


Figure 4.2: Plot of pairwise similarities (measured by FSIM) of frames from the video *America's New Frontier, Segment 10*, along with the detected shot boundaries (using the threshold $T_B = 0.1$).

Once the shot boundaries are detected, the keyframe extraction process is started. This process picks one representative frame for each shot. Let K be the total number of shots, which is the number of detected boundaries plus one. Also, let $begin_k$ and end_k be, respectively, the indices of the beginning and the ending frames of the shot k, with $k = 1, 2, ..., K$. Taking advantage of the similarity matrix S calculated at the second stage, the representative frame of shot k, denoted by RF_k, is defined as:

$$\text{RF}_k = \{begin_k \leq f \leq end_k \mid \max \sum_{g=begin_k}^{end_k} S(f,g)^2\} \tag{4.1}$$

In other words, Equation 4.1 selects the frame f that is the most similar to the other ones of the same shot, which corresponds to the line of the similarity submatrix related to shot k whose FSIM_C values are usually very high. This approach works very well for

videos with none or little movement, where the FSIM_C values between consecutive frames are generally close to 1 and do not oscillate too much inside each shot.

Once all the representative frames are obtained, a redundancy elimination algorithm is run to eliminate similar keyframes. From S, the FSIM_C's between all pairs of representative frames are observed, and if a value is equal to or greater than a similarity threshold T_S, a frame is discarded. For this process, T_S was fixed at 0.75. Then, the remaining frames will make part of the final summary.

Furthermore, a fundamental improvement over the previous approach lies on the usage of an objective metric of evaluation, instead of using subjective methods. For this purpose, the CUS (Custom User Summaries) metric, proposed by Avila et al. [38], was used. The authors evaluated their summaries by using their own features used to describe the video frames, which include selecting a specific number of bins of the hue band of the HSV color histogram. On the other hand, Mahmoud et al. [89] modified the CUS metric so that it could evaluate the summaries according to their descriptor (combination of color and texture features). Therefore, the evaluation will be based on the FSIM_C metric.

The advantage of VSQUAL resides on the fact that the video frames are represented by a more objective measure, which reflects the way that humans perceive images. Moreover, comparing to our previous approach [31], VSQUAL does not employ clustering algorithms, extracting keyframes in a simpler way, and it can be easily adapted to any other image quality metric. However, this new approach still has some problems in dealing with videos that have considerable movement, which causes oscillations at the FSIM_C's between consecutive frames, thus being harder to find boundaries (especially at gradual transitions) and more likely to select different keyframes from other approaches' summaries, thus reducing the CUS scores.

4.3 Experimental Results

Like the approach described in Chapter 3, the VSQUAL approach was tested with the OVP database, along with a modified version of the CUS metric to evaluate the summaries. The concept of frame similarity is the same used in the redundancy elimination step described in Section 4.2, where two frames are considered similar if the FSIM_C value between them is equal to or greater than $T_S = 0.75$.

Taking the summaries provided by the authors of VSUMM [38] as the ground-truth, the evaluation is based on three different values: number of similar frames SF_i (i.e., frames from an automatic summary that matched to frames from user summaries), number of frames in the automatic summaries AS_i and number of frames in the user summaries US_i, where $i \in \{1, 2, 3, 4, 5\}$ represents a specific user. From these values, both precision P_i and recall R_i values can be obtained, with $P_i = SF_i \ / \ AS_i$ and $R_i = SF_i \ / \ US_i$. Since there is a trade-off between precision and recall [7], the F-measure is used as the quality assessment metric for the automatic summaries. For each video, the F-measure is defined as the average of the five F-measure values obtained from each user, as stated in

Equation 4.2:

$$\text{F-measure} = \frac{\sum_{i=1}^{5} \frac{2 \times P_i \times R_i}{P_i + R_i}}{5} \tag{4.2}$$

Table 5.1 shows the F-measures for the summaries of some videos from the database for each method, along with the average using the entire database. From the table, it can be seen that VSQUAL performs better than most of the methods of the literature, also being comparable to recent approaches such as VSUMM and VSCAN, even though it is behind VSUMM on the total average. Despite the good results, the proposed method still needs some adjustments, once using only the FSIM_C values for detecting similarities leads to several false negatives, thus producing low F-measure values in some cases.

Table 4.1: F-measures of the summaries produced by each method and video.

Video	VSQUAL	DT	OVP	STIMO	VSUMM	VSCAN
The Great Web of Water (Seg. 2)	0.745	0.198	0.707	0.680	**0.840**	0.527
A New Horizon (Seg. 4)	**0.411**	0.195	0.122	0.181	0.335	0.340
Senses And Sensitivity (Lect. 2)	0.800	0.444	0.667	0.625	**1.000**	0.727
Exotic Terrane (Seg. 3)	**0.691**	0.449	0.521	0.370	0.580	0.624
America's New Frontier (Seg. 1)	**0.622**	0.413	0.476	0.301	0.572	0.599
America's New Frontier (Seg. 10)	0.414	0.268	**0.526**	0.443	0.470	0.499
Oceanfloor Legacy (Seg. 2)	**0.653**	0.470	0.599	0.365	0.644	0.636
Hurricane Force (Seg. 3)	0.615	0.545	0.548	0.366	**0.692**	0.570
Drift Ice (Seg. 6)	**0.964**	**0.964**	0.766	0.749	**0.964**	0.916
Drift Ice (Seg. 7)	**0.856**	0.713	0.763	0.616	0.814	0.846
Overall Average (50 videos)	0.481	0.361	0.470	0.401	**0.511**	0.479

Figures 4.3 and 4.4 show the results of the proposed method, comparing the F-measures against other methods of the literature. Summaries that were manually created by 5 different users are also available as the ground-truth. Looking at the results for the first video, it can be seen that VSQUAL managed to cover most of the content selected by the users, in spite of producing a lower F-measure than VSUMM. However, both values could have been much closer if the comparison between the first frame of VSQUAL summary and the similar frames from the user summaries were not regarded as a mismatch, according to the FSIM_C metric, which evidences the major drawback of the distance metric: the variance to translations.

For the second video, except for the third frame of user 1, which is also present in the other users' summaries and in most of the automatic summaries, VSQUAL also covered the whole content of the video. Despite that difference, VSQUAL was able to surpass all of the other methods, especially for the fact that the number of frames contained in the summary is very close to the average of the users, which is the main reason that VSCAN could not achieve a better result. In addition, there were no false positives / negatives when using the FSIM_C for detecting image similarities.

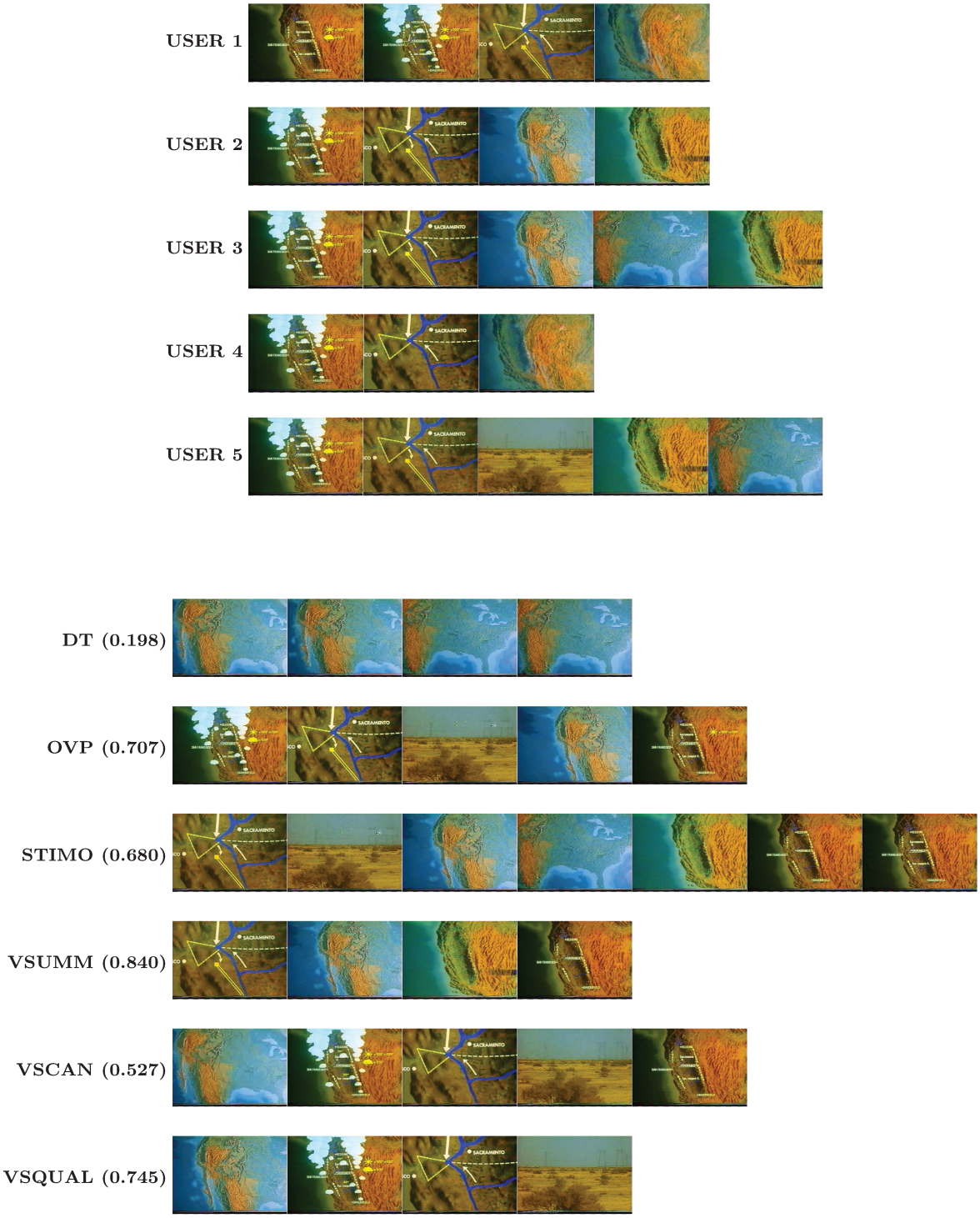


Figure 4.3: Summarization results for *The Great Web of Water - Segment 02*, along with the respective F-measures.

4.4 Discussion

This chapter described a video summarization method using objective IQA metrics, which are developed in such a way that is close to the way that humans perceive image qualities. FSIM_C was chosen to describe the video frames. The methodology pipeline was strictly based on the construction of the FSIM_C similarity matrix, from which it was possible to

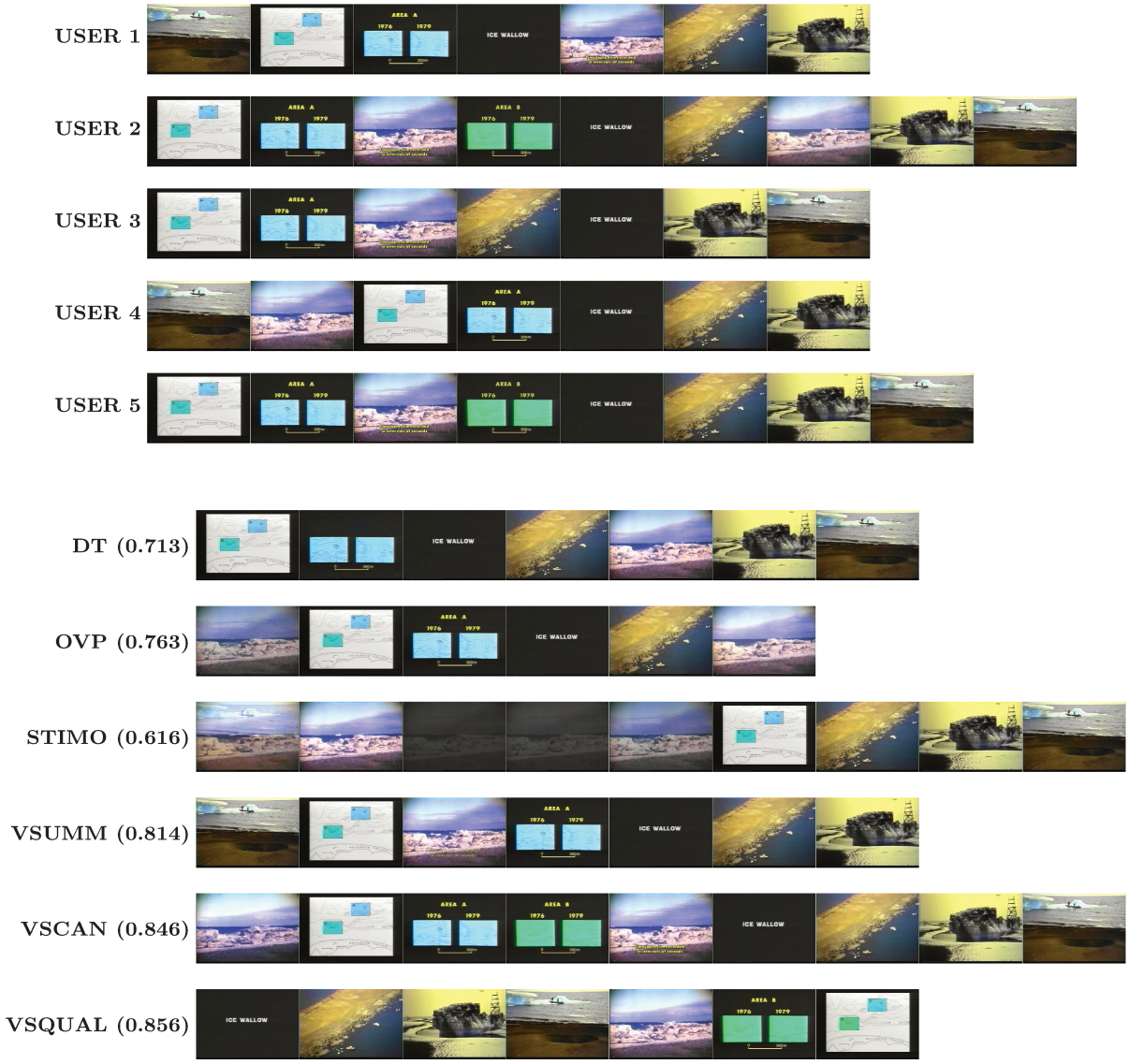


Figure 4.4: Summarization results for *Drift Ice as a Geologic Agent, Segment 07*, along with the respective F-measures.

execute three different algorithms: shot boundary detection, keyframe extraction and redundant frame elimination. The final summary is created after those steps and compared to summaries produced by different video summarization approaches through the CUS metric.

Results showed that the proposed method is comparable to the state-of-the-art techniques, even though improvements are needed in order to increase the accuracy in the similarity detection process. These improvements include: combination of FSIM_C with other feature(s), such as motion and spatio-temporal features to overcome the problem of dealing with contents with intense movement; a further analysis of the shot boundary detection step; and development of other algorithms for choosing the representative frame of each shot, with the goal of producing summaries that are as close as possible to the way that humans do.

In addition, the subjectivity factor in the result analysis was vastly eliminated. How-

ever, the similarity measure was not very effective because of the drawbacks of the FSIM_C, not only regarding the existence of several false negatives but also the general performance. Therefore, the following method will focus on resolving those issues.

Chapter 5

Video Summarization by Color Co-occurrence Matrices

This chapter presents an approach to video summarization named VISCOM (**VI**deo **S**ummarization by Color **C**o-**O**ccurrence **M**atrices), which extends the method described in Chapter 4 by using color co-occurrence matrices as the image descriptor for video frames. General concepts used in this work, along with the proposed methodology, are detailed, as well as the improvements related to the previous approach and a detailed result analysis.

5.1 Co-Occurrence Matrices

A common approach to image description is to use histograms, due to its easy and efficient implementation at the same time it provides an useful statistical information about a content present in an image. Nonetheless, they are highly dependent on the color distribution of that content, ignoring other attributes such as shape, texture and spatial localization. Additionally, it is important to choose a set of attributes for describing an image that best preserves the computational efficiency, which depends directly on the application domain.

Another popular approach is the use of texture descriptors. One important step in constructing these descriptors is that they must convey information about the relative positions of the pixels with respect to each other [52]. Co-occurrence matrices are a very suitable structure for accomplishing this task, since they do not only consider the color distribution of pixels (just like histograms do) but also their spatial information.

The construction of co-occurrence matrices goes as follows: given an image I of size $W \times H$ with L possible values for pixel intensity (in case of a grayscale image), construct the matrix P of size $L \times L$ such that each element $P(i, j \mid t)$ (where $t = (d, \theta)$ is a translation vector with distance d and direction θ) of the matrix denotes the probability that a pixel with intensity value j occurs at direction θ and distance d in relation to a pixel with intensity value i in the image I. In other words, let $p = (x, y)$, where $0 \leq x \leq H - 1$ and $0 \leq y \leq W - 1$, be a pixel in I and $q = (\Delta x, \Delta y)$, with $\Delta x = x + d \cos\theta$ and $\Delta y = y + d \sin\theta$ a translation of p, such that q remains in the spatial domain of I. The

computation of each element of P is done according to Equation 5.1:

$$P(i,j \mid t) = \text{card}\,\{\{p,q\} \in I \mid I(p) = i, I(q) = j\} \tag{5.1}$$

Figure 5.1 illustrates an example of how these matrices are constructed. Matrix P is also known as Gray Level Co-occurrence Matrix (GLCM) [58, 126]. It has been widely used to extract texture features, such as energy, contrast, entropy, homogeneity and inverse difference. In a similar way, Color Co-occurrence Matrices (CCM) [10, 64, 75] can be used to represent the distribution of color features between pairs of pixels in an image, considering the correlations between the color bands as well.

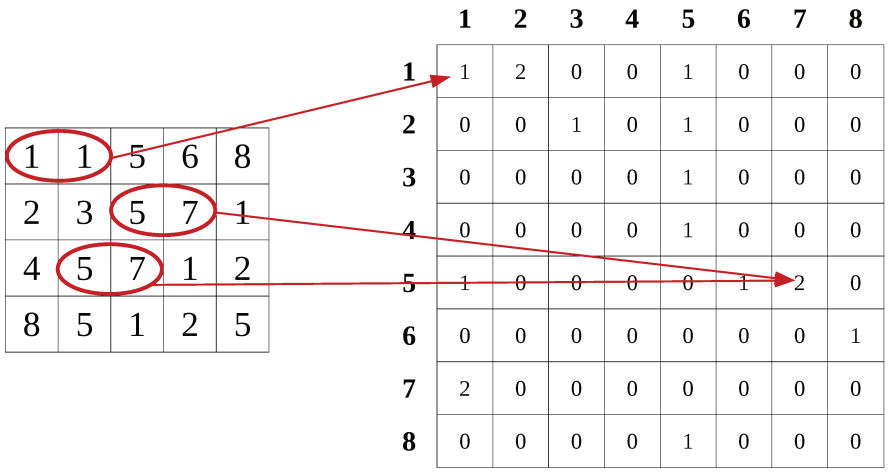


Figure 5.1: Gray-level co-occurrence matrix computed from an image quantized in $L = 8$ levels of pixel intensity.

The construction of the CCM's from a multispectral image I proceeds as follows: let $C_1, C_2, ..., C_n$ be the n channels of I, where each one is coded on L levels, and L the number of rows and columns of the CCM's. Also, let C_u and C_v be a pair of channels (with $1 \leq u, v \leq n$). Finally, considering the same points p and q as described before, the computation of each position (i, j) of the CCM of size $L \times L$ and a translation vector t for a pair of channels C_u and C_v is done according to Equation 5.2:

$$\text{CCM}_{(C_u,C_v)}(i,j \mid t) = \text{card}\,\{\{p,q\} \in I \mid C_u(p) = i, C_v(q) = j\} \tag{5.2}$$

This procedure results in a set of matrices, representing all combinations of pairs of channels. Like color histograms, CCM's can be extracted from any color system (RGB, HSV, YCbCr, and so on). Figure 5.2 shows an example for the RGB color space.

In this work, the RGB color space was used to represent the video frames, with $L = 8$

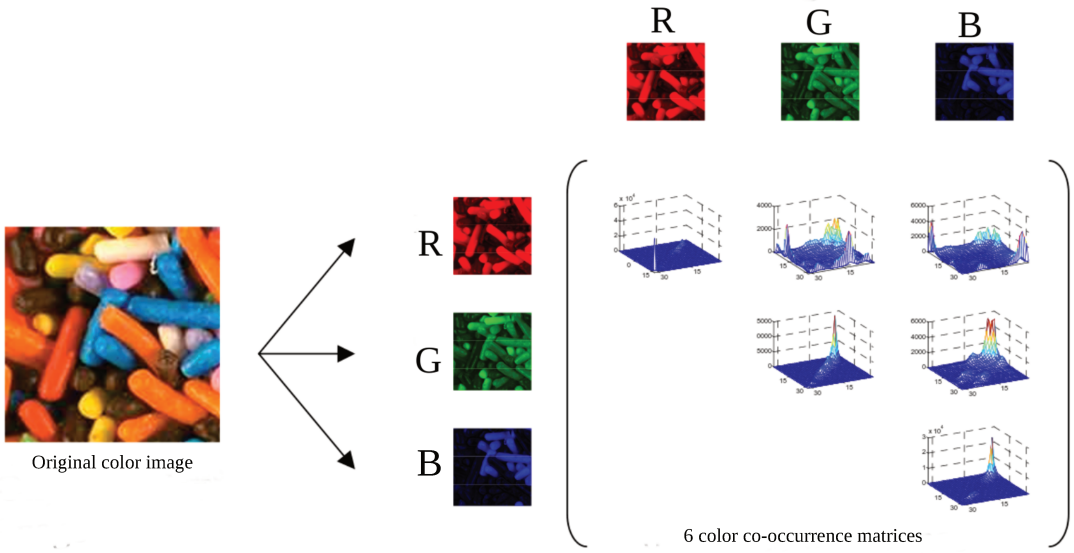


Figure 5.2: Color co-occurrence matrices extracted from a image in the RGB color space. Image extracted from [10].

and $t = (1, 0)$ (one pixel to the right). Since $\text{CCM}_{(C_u,C_v)}(i, j \mid t)$ and $\text{CCM}_{(C_v,C_u)}(i, j \mid t)$ store the same information, there are only 6 possible pairs of channels (C_u, C_v), thus leading to 6 different CCM's: (R,R), (R,G), (R,B), (G,G), (G,B) and (B,B).

5.2 Template Matching and Distance Functions

Template matching is a recurrent problem in image processing, computer vision and pattern recognition that looks for areas in an image that are similar to a template subimage. One of the techniques used for this procedure is the digital image correlation [18, 61, 101], which measures the displacement between a pair of images or between an image and a template subimage by means of a similarity / dissimilarity function. In turn, this function computes the displacement field that corresponds to the best possible match between the input images.

The concept can also be extended to the 3-D space, but only the 2-D image context will be analyzed, henceforth. Let I be an image of size $W \times H$ and T a template image of size $w \times h$ which is desired to be found somewhere in I. The displacement field is represented by a matrix DF of size $(W - w - 1) \times (H - h - 1)$ where each position stores the comparison between T and an overlapped region of I that has the same size of T. This comparison can be done in several ways, by means of a specific similarity function, according to Equation 5.3:

$$\text{DF}(i, j) = \text{FUNC}\ (T(i', j'), I(i + i', j + j')) \tag{5.3}$$

where FUNC represents the similarity function. It is worth mentioning that if I and T have the same size, DF will be composed by a single number. Therefore, when applying

Equation 5.3 taking two images of the same size as an input, the result can be interpreted as the similarity / dissimilarity degree between those images.

One of most popular functions used is the Cross-Correlation (CC) [80], as defined in Equation 5.4:

$$\mathrm{CC}(i,j) = \sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}(T(i',j') \cdot I(i+i', j+j')) \tag{5.4}$$

However, besides the fact that $\mathrm{CC}(i,j)$ is dependent on the size of the feature used to describe the input images, CC has the drawback of not being invariant to changes in image amplitude (e.g., brightness and contrast variations) [80]. This problem can be solved by normalizing the image and feature vectors so that they have an unitary length, yielding a new function called Normalized Cross-Correlation (NCC) [44], as it can be seen in Equation 5.5:

$$\mathrm{NCC}(i,j) = \frac{\sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}(T(i',j') \cdot I(i+i', j+j'))}{\sqrt{\sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}T(i',j')^2 \cdot \sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}I(i+i', j+j')^2}} \tag{5.5}$$

An even more robust version is the Zero-Mean Normalized Cross-Correlation (ZNCC) [44], which is also offset invariant and it is calculated by subtracting every term of Equation 5.5 by the mean intensity value of I or T, according to Equation 5.6:

$$\mathrm{ZNCC}(i,j) = \frac{\sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}((T(i',j') - \mu(T)) \cdot (I(i+i', j+j') - \mu(I)))}{\sqrt{\sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}(T(i',j') - \mu(T))^2 \cdot \sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}(I(i+i', j+j') - \mu(I))^2}} \tag{5.6}$$

Another similar function is the Sum of Squared Differences (SSD), which also has its Normalized (NSSD) and Zero-Mean (ZNSSD) versions [102]. Their definitions are stated in Equations 5.7, 5.8 and 5.9, respectively:

$$\mathrm{SSD}(i,j) = \sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}(T(i',j') - I(i+i', j+j'))^2 \tag{5.7}$$

$$\mathrm{NSSD}(i,j) = \frac{\sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}(T(i',j') - I(i+i', j+j'))^2}{\sqrt{\sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}T(i',j')^2 \cdot \sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}I(i+i', j+j')^2}} \tag{5.8}$$

$$\text{ZNSSD}(i,j) = \frac{\sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}((T(i',j') - \mu(T)) - (I(i+i',j+j') - \mu(I)))^2}{\sqrt{\sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}(T(i',j') - \mu(T))^2 \cdot \sum_{i'=0}^{w-1}\sum_{j'=0}^{h-1}(I(i+i',j+j') - \mu(I))^2}} \tag{5.9}$$

Among the described functions, CC, NCC and ZNCC are used as similarity measures whereas SSD, NSSD and ZNSSD are used for calculating dissimilarity. According to Pan et al. [102], there is an equivalence between ZNCC and ZNSSD. The only difference is their value ranges, where the former ranges from -1 to 1 and the latter from 0 to 4. Both NCC and NSSD range from 0 to 1, in spite of their ambivalences.

In order to increase the computational efficiency, co-occurrence matrices can be used as the input for the similarity function, rather than using the original image. In the case of color matrices, taking the RGB color space as an example, since there are six co-occurrence matrices (representing a pair of color bands) for each input image, the similarity function computes six values, one for each pair of corresponding matrices, i.e., the similarity between the (R,R) matrix of I and the one from T, then between (R,G) matrices, followed by (R,B), and so on. The final value can be expressed by some kind of mean between all values (arithmetic mean, weighted mean, harmonic mean, etc.).

5.3 Proposed Methodology

The methodology proposed in this work is an improved version of [32], described in Chapter 4, which used an objective image quality assessment metric to compare pairs of video frames. Even though this metric showed some efficiency, a substantial amount of time is required to measure the distances between all pairs of frames. To overcome this problem, the CCM descriptor, described in Section 5.1, was used to represent the video frames.

Figure 5.3 shows an overview of VISCOM. At the first stage, the frames are sampled in a smaller amount in order to save some computational time for the whole summarization process, at the same time it does not discard any piece of meaningful information. In this work, a sampling of 15 frames per second was used, which produced slightly better results than other tested values (5, 10 and 30 frames per second).

After that, the CCM descriptor is computed for each sampled frame. Then, if there are monochromatic frames in the frame set, they are discarded so that they are not included in the final summary. This is done by calculating the entropy of all frames, using their respective CCM's. A frame is discarded when the entropies of all of its CCM's equal zero, meaning that the frame has a uniform color distribution across the frame.

At the next stage, using the remaining frames from the previous process, the distances between consecutive frames are computed. This process is done by using the NSSD as the distance function (defined in Equation 5.8), which was proved to be very robust and widely used in tasks that deal with digital image correlation. Since each frame is represented by six distinct matrices, the distance between two frames is calculated by taking the average

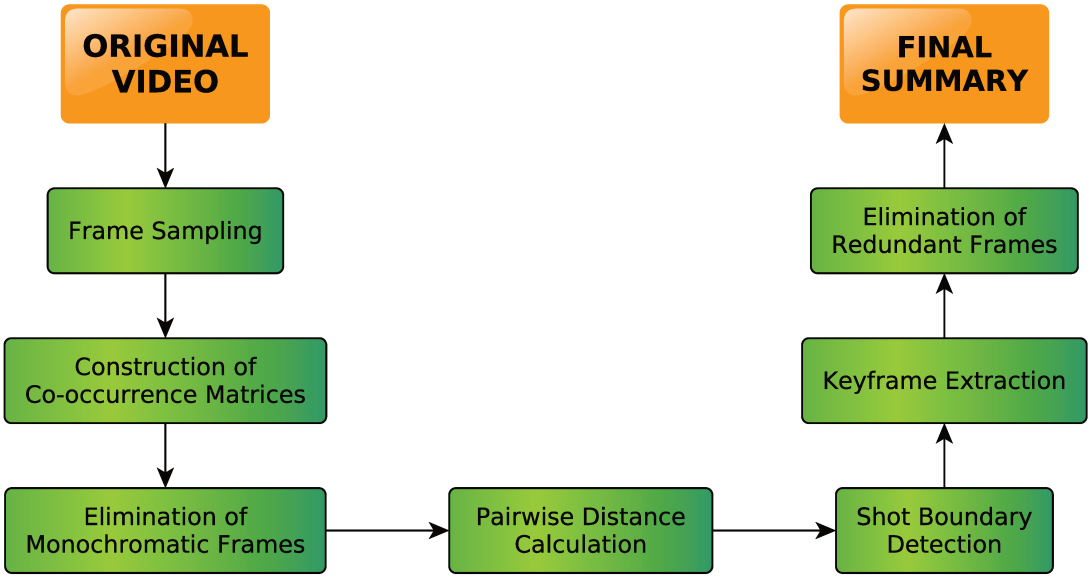


Figure 5.3: Overview of the stages of VISCOM method.

of the NSSD's between the corresponding co-occurrence matrices, which can be obtained according to Equation 5.10:

$$\mathrm{D}(I, J) = \frac{\sum_{c=1}^{6} \mathrm{NSSD}(I_c, J_c)}{6} \tag{5.10}$$

where c is one of the six co-occurrence matrices described in Section 5.1, where I_c and J_c are two input images expressed in terms of their co-occurrence matrices. Since NSSD ranges from 0 to 1, Equation 5.10 has the same range, where the closer to zero, the more similar the images are.

Once the pairwise distances of all frames are calculated, the shot boundary detection process is started. The process is done exactly as described in Section 4.2, except for the fact that the SBD algorithm looks for points of local maxima in a sequence of pairwise distances, since a distance function is used. Comparing this plot to the one from Figure 4.2, it can be noticed that the NSSD provides a much more stable variation of the distances during a shot than the FSIM version and the peaks are much easier to be identified. Moreover, the threshold values for this process were defined at $T_B = 0.075$ and $m = 9$, by means of empirical tests. Figure 5.4 shows an example for a given video. From the plot, it can be inferred that intervals of low values represent frames that belong to a common shot, which happens because consecutive frames of a shot generally have similar contents (i.e., low NSSD values). Moreover, whenever there is an abrupt increasing in the NSSD values, it means that a shot transition may be detected, since the contents of a different shot are about to appear at a video. Then, the values become lower again, representing the frames of a new shot.

With the detected shots, the keyframes are then extracted, which is done by choosing one single frame to represent each respective shot. Unlike the previous method, which

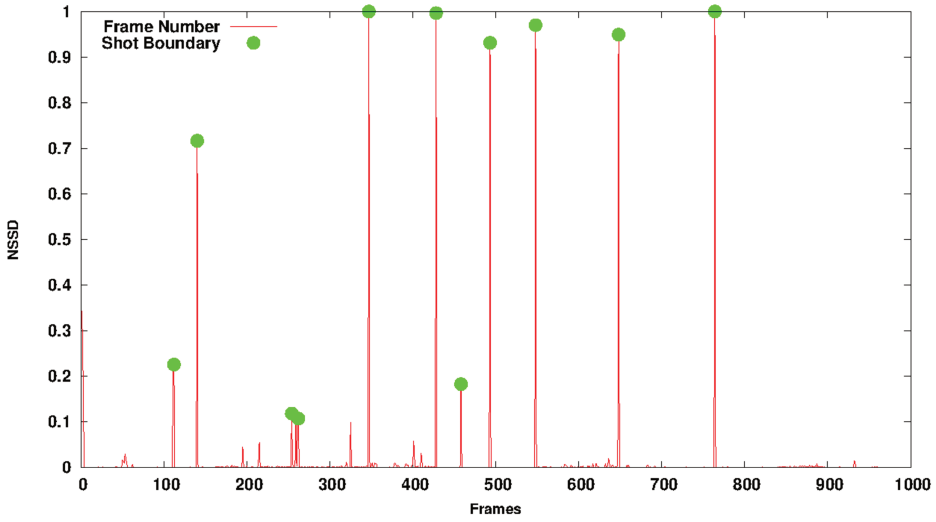


Figure 5.4: Plot of pairwise distances of frames from the video *America's New Frontier, Segment 10*, along with the detected shot boundaries (using the threshold $T_B = 0.075$).

took a substantial amount of time to choose the representative frames, the middle core frame in sequential order of a shot is often a good choice for this task, especially due to performance optimization reasons. However, shots that have little content are discarded, since they may not be relevant for the final summary. So, if the number of frames of a shot is below a threshold T_K, the shot is discarded and, therefore, a keyframe is not selected for that shot. This threshold is defined by a percentage of the total number of remaining frames after the elimination of monochromatic frames. A value of $T_K = 2\%$ was enough to produce satisfactory results.

Later, a redundancy elimination algorithm is executed to discard similar keyframes. Here, the distances of all pairs of keyframes are analyzed. If a distance of a given pair of keyframes is above a distance threshold $T_D = 0.2$, one of these keyframes is discarded. Finally, the remaining frames comprise the final summary, which is then compared against several methods of the literature by means of the CUS metric, describing all frames using the same CCM descriptor and the NSSD as the distance metric.

The advantage of VISCOM lies on the robustness and effectiveness of both the image descriptor and the distance function. On the other hand, regarding the identification of similarities in images, it can still find some false positives and negatives, in a sense that pairs of images that have different contents but similar color distributions may lead to low values for the distance function or vice-versa. Examples of these behaviors are shown in Figures 5.5 and 5.6, respectively.

5.4 Experimental Results

The tests were conducted on an AMD FX-6300 3.5 GHz processor and 4 GB of memory. The summarization method described in Section 5.3 was implemented with the OpenCV platform [2]. Like the other two approaches, the OVP database was used in the experiments. Results obtained by VISCOM were then compared against different video

Figure 5.5: Frames from video *The Great Web of Water, segment 01* with different contents, but with a NSSD of 0.0417396 (false positive).

Figure 5.6: Frames from video *The Great Web of Water, segment 01* with similar contents, but with a NSSD of 0.575964 (false negative).

summarization methods, which include: Delaunay Triangulation (DT) [93], STIMO (STIll and MOving Video Storyboards) [50], VISON (VIdeo Summarization for ONline applications) [7], VSCAN (a summarization method that uses the DBSCAN clustering algorithm) [89], our previous approach VSQUAL [32], as well as the OVP summaries and the ones provided by VSUMM [38].

In order to measure the performance of VISCOM, the implementation was executed 10 times with all videos from the database. The average execution time was 606.7 ± 2.8 seconds (about 12 seconds per video). In addition, the individual execution times for each video varied between 5 and 35 seconds. As an overall results, these times are very satisfactory, once each summary is generated in a small percentage of the total time of each respective video (usually between 10% and 20%). Such performance is affected not only by the video frame count, but also by the number of extracted keyframes, because the higher this number is, more comparisons are made during the redundancy elimination stage.

Table 5.1 shows the average precisions, recalls and F-measures for the summaries of all videos from the database for each method. It can be seen, from the table, that VISCOM overcomes the evaluated state-of-the-art approaches, producing competitive results while maintaining a good trade-off between speed and quality.

Furthermore, a variation of VISCOM using spectral clustering was also tested. In this variation, the keyframe extraction step is done similarly to the method described

Table 5.1: Average precision, recall and F-measures of the summaries produced by each method for the entire database.

Method	Precision (%)	Recall (%)	F-Measure
DT [93]	54.7	43.3	0.469
STIMO [50]	51.9	62.1	0.552
OVP [3]	58.4	65.7	0.589
VSQUAL [32]	55.7	74.3	0.608
VISON [7]	59.5	67.5	0.619
VSUMM [5]	72.1	64.1	0.666
VSCAN [89]	62.5	83.1	0.702
VISCOM	64.9	81.1	0.706

in Section 3.3, but in this case, the affinity matrix is generated using the NSSD as the distance function. This approach obtained an average F-measure of 0.696, with precision and recall rates of 63.1% and 80.8%, respectively, which outperforms most of the methods available in the literature, except for VSCAN and the original VISCOM approach as well. Without the spectral clustering and applying K-means directly on the affinity matrix, the average F-measure obtained was 0.615, with 53.8% of precision and 74% of recall.

It is important to highlight that the CUS scores are strongly dependent on the image similarity function used and how the images are described, because a similarity (between a pair of frames) detected by a specific function may not be detected by another. Thus, each video summarization method is evaluated using its own descriptors and similarity functions, along with a proper similarity threshold, as done in VSUMM, VISON and VSCAN, where each approach produces different average F-measures, including for DT, OVP and STIMO approaches. By observing the results for VSQUAL, it can be noticed that even though it had a good performance by using FSIM as the similarity function, the same cannot be said when using the NSSD function and the descriptor proposed for VISCOM.

Figures 5.7, 5.8 and 5.9 show the results of VISCOM for some specific videos, along with the respective results generated from other summarization methods and their respective F-measures, as well as the summaries that were manually made by 5 different users. For the first example, even though VISCOM did not achieve the highest F-measure among the tested approaches, it could still manage to cover a great amount of the aspects of the video content, according to the ground-truth. It is also important to notice the presence of the third and fourth frames of the second row in the final summary. Looking at the images, it can be seen they both belong to the same shot, but the redundancy elimination algorithm failed at discarding one of them (which also happened with VSCAN and VSQUAL). Despite the little difference in the content between these images, the distance between them, according to the distance metric used, was 0.475, which is above T_D.

Concerning the second example, the automatic summaries covered all the content selected by the users. However, the key difference between the summaries lies on the size of the produced summaries related to the average size of the user summaries. Both VISCOM and VISON performed very well in this task, but the former was a slightly better in terms

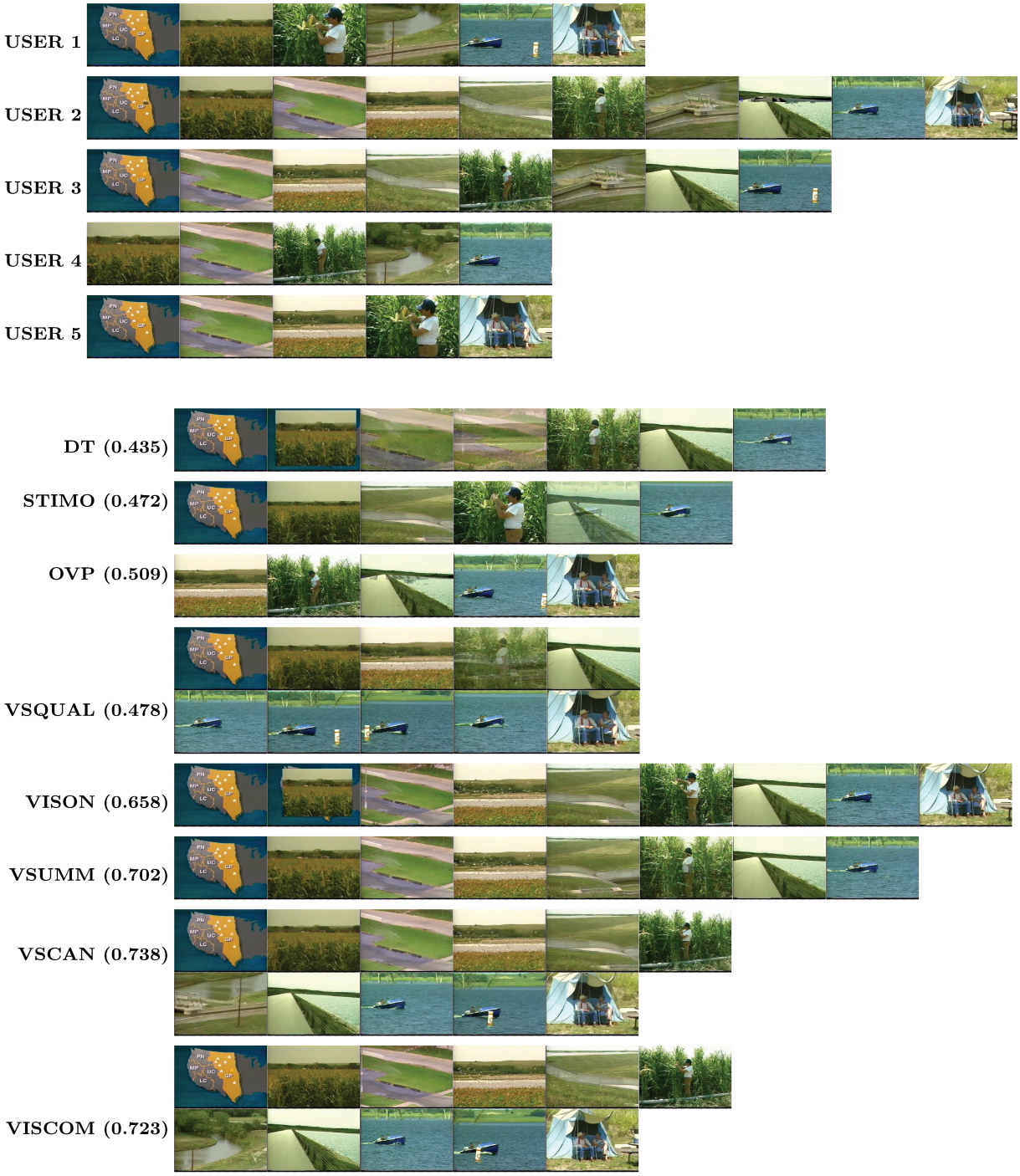


Figure 5.7: User summaries and automatic summaries of each method from the video *A New Horizon, Segment 06*, along with the respective F-measures.

of F-measure. The other approaches either managed to cover the whole content, but in expense of a higher size of the final summary (STIMO, OVP and VSCAN), or produced shorter summaries which a less satisfactory content (DT, VSQUAL and VSUMM). The similar happens in the third example, with VISCOM, VSUMM and OVP achieving the best results, but the other approaches obtained worse results because of the size of their respective summaries.

Another point that must be taken into account refers to the strategy for selecting

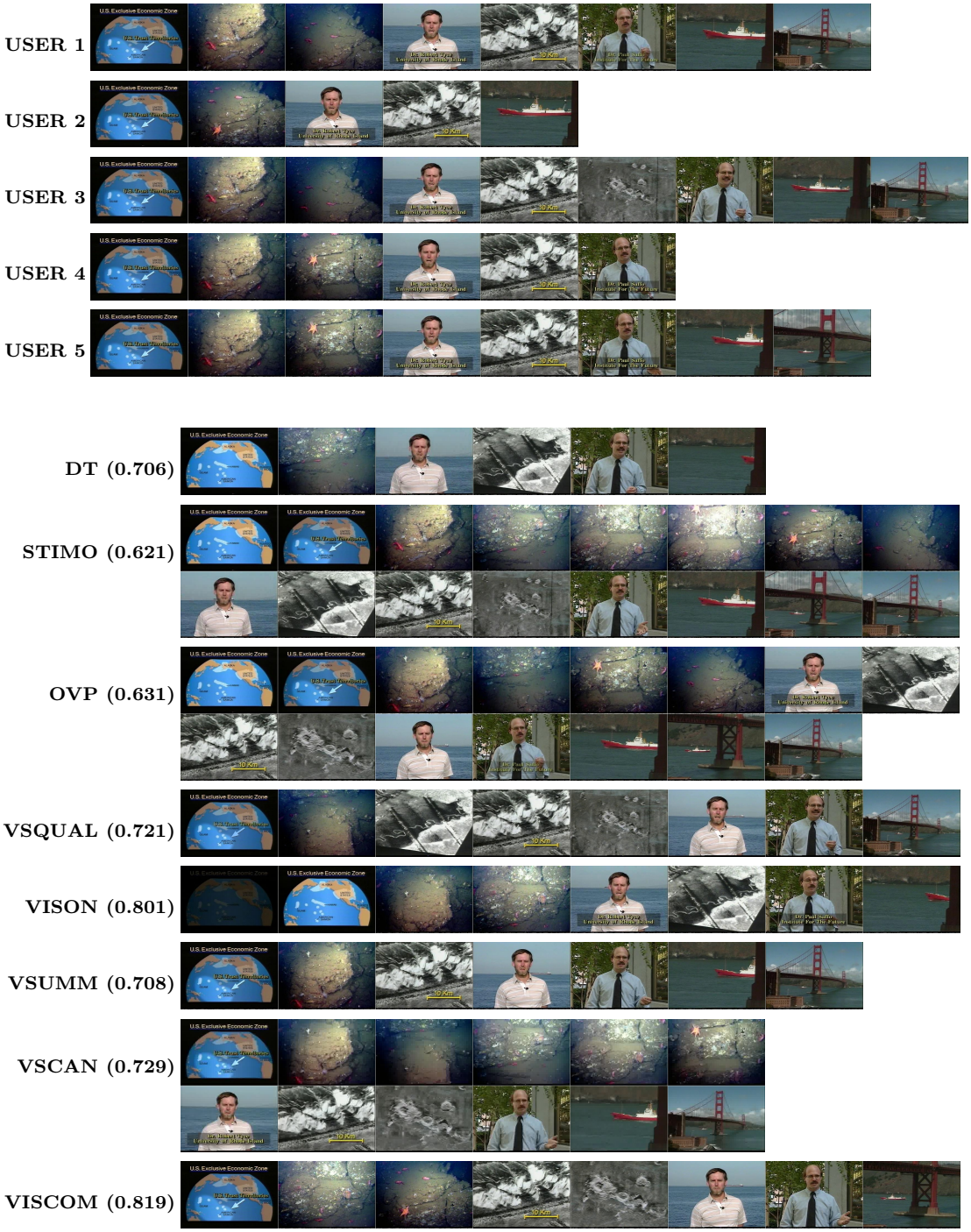


Figure 5.8: User summaries and automatic summaries of each method from the video *America's New Frontier, Segment 10*, along with the respective F-measures.

the representative frame for a shot. VISCOM chooses the middle core frame of the detected shots (like VISON and VSCAN), whereas VSUMM, for example, uses clustering algorithms to group the frames in shots (clusters), selecting the ones that are closest to the centroids of each cluster as keyframes. As stated in Section 5.3, depending on the chosen frame, the distance metric may not consider a pair of frames as similar (generating false negatives), regardless of having the same background and objects. The opposite (false positives) also happens, but much less frequently than the other case. Therefore, both

cases have a direct impact on the F-measure scores. Since false negatives occur more often for the chosen distance metric and image descriptor, the F-measure scores for all approaches tend to be a little bit lower than expected.

Despite the aforementioned issues, the distance function is very helpful in the task of identifying image similarities for the absolute majority of the cases. In addition, the keyframe selection strategy used in VISCOM is very suitable for generating satisfactory summaries that reflect the humans' concept of importance. Any change in this strategy might cause a significant increase in the computational time, at the same time it is not worth the eventual gain on the average F-measure, since the visual changes in the summaries are little.

5.5 Discussion

This chapter described VISCOM, an approach to video summarization that uses color co-occurrence matrices in the RGB color space as the image descriptor for the frames. The NSSD was defined as the distance function for detecting image similarities, which was shown to be very effective in most of the cases.

Considering the results of spectral clustering and the proposed method, one may think that the use of clustering algorithms is unnecessary, since the shot boundary detection step already provides a division of the frames into groups, i.e., the video shots. However, this assumption is insufficient to completely discard those algorithms, because it might be possible that better results can be achieved by finding an optimal parameterization.

The main contributions of this work include: improvement over the previous approach (VSQUAL, described in Chapter 4) and other methods of the literature, development of a fast and robust algorithm, which can be integrated to platforms that deal with video indexing and retrieval, and an extension of the ground-truth for automatic summaries so that other approaches can use it for objective comparisons.

Figure 5.9: User summaries and automatic summaries of each method from the video *Exotic Terrane, Segment 04*, along with the respective F-measures.

Chapter 6

Conclusions

This thesis investigated the field of video summarization, providing a general contextualization, as well as an analysis of different methods of the literature in terms of knowledge domain, strategies for each stage of the summary creation process and metrics that assess the quality of the summaries. In addition, three different approaches were proposed, differing in several aspects, such as how the video frames were described, how the keyframes were selected to compose the final summary and the metrics used to identify image similarities when comparing to a ground-truth.

In general, the summaries generated by each method covered most aspects of the contents of each video used in the tests, thus producing satisfying results. However, as the approaches were evolving, there was a need for defining more robust image descriptors, similarity functions and objective evaluation metrics in order to improve the quality of the results, as it could be observed in the transition from the first proposed approach to the second one. On the last approach, those objectives could be finally achieved, along with results with a superior quality than some other methods of the literature, according to the CUS evaluation metric. Hence, among the three methods presented in this work, the third one is the most recommended for a possible integration in several kinds of applications that deal with video indexing, retrieval and content analysis, especially due to its general performance.

Among the perspectives for future work, one of them is related to the analysis of different features and descriptors that can be suitable for improving the quality of the results, at the same time they grant a high accuracy on the task of detecting image similarities. Some suggestions include, but not limited to, bag-of-visual words [138], spatio-temporal features [74] and some variations of the color co-occurrence matrix descriptor proposed in Chapter 5. Hence, a standardization of the CUS metric can be achieved so that every new summarization method can have a more confident evaluation, rather than using its own descriptors and similarity functions for that purpose.

Other strategies for choosing important events of a video and keyframe extraction can also be taken into account, such as analysis of motion features, object tracking and image aesthetics, or even an in-depth analysis of some of the ones mentioned in this thesis, such as the use of a spectral clustering algorithm, which was shown, on the last proposed approach, that it can generate results that are as good as the best methods evaluated in that work. A possible alternative is the change of K-means for an image clustering

algorithm [72].

Eventually, it is desirable to increment the ground-truth with several other videos, along with some users' opinions as the main reference, including the addition of other genres, higher quality videos and unstructured videos (especially home videos), so that the comparison of a video summarization method can be extended to more methods and more types of videos. Then, a general framework for evaluation of video summaries that uses CUS as the quality metric can be developed, establishing specific guidelines for the evaluation of summaries, such as the TRECVID database, mentioned in Section 2.1.4.

Given the fact that subjective features are usually difficult to model, as it happens with the concept of important events of a video, there is still room for improvements on the video summarization field. Although an "ultimate consensus" may never be attained, recent advances have corroborated the constant evolution in the simulation of the human behavior, not only in the video summarization context, but also in other fields, including artificial intelligence, computer vision and machine learning. In this way, several possibilities can be considered toward a concise way of analyzing all kinds of videos.

Bibliography

[1] MATLAB, 2015. http://www.mathworks.com/products/matlab/.

[2] OpenCV: Open Source Computer Vision, 2015. http://www.opencv.org.

[3] The Open Video Project, 2015. http://www.open-video.org.

[4] TREC Video Retrieval Evaluation: TRECVID, 2015. http://trecvid.nist.gov.

[5] VSUMM (Video SUMMarization), 2015. https://sites.google.com/site/vsummsite.

[6] YouTube, 2015. https://www.youtube.com/yt/press/statistics.html.

[7] J. Almeida, N. J. Leite, and R. S. Torres. VISON: VIdeo Summarization for ONline Applications. *Pattern Recognition Letters*, 33(4):397–409, Mar. 2012.

[8] A. M. Amel, B. A. Abdessalem, and M. Abdellatif. Video Shot Boundary Detection Using Motion Activity Descriptor. *Journal of Telecommunications*, 2(1):54–59, Apr. 2010.

[9] A. Aner and J. R. Kender. Video Summaries Through Mosaic-Based Shot and Scene Clustering. In *7th European Conference on Computer Vision-Part IV*, pages 388–402, London, UK, 2002. Springer-Verlag.

[10] V. Arvis, C. Debain, M. Berducat, and A. Benassi. Generalization of the Coocurrence Matrix for Colour Images: Application to Colour Texture Classification. *Image Analysis & Stereology*, 23(1):63–72, 2011.

[11] J. Baber, N. Afzulpurkar, M. Dailey, and M. Bakhtyar. Shot Boundary Detection From Videos Using Entropy and Local Descriptor. In *17th International Conference on Digital Signal Processing*, pages 1–6, July 2011.

[12] W. Bailer, E. Dumont, S. Essid, and B. Mérialdo. A Collaborative Approach to Automatic Rushes Video Summarization. In *International Conference on Image Processing*, pages 29–32, San Diego, California, USA, Oct. 2008.

[13] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *9th European Conference on Computer Vision*, pages 404–417, May 2006.

[14] S. Benini, P. Migliorati, and R. Leonardi. Hidden Markov Models for Video Skim Generation. In *Eight International Workshop on Image Analysis for Multimedia Interactive Services*, pages 6–9, Washington, DC, USA, 2007. IEEE Computer Society.

[15] V. Benni, R. Dinesh, P. Punitha, and V. Rao. Keyframe Extraction and Shot Boundary Detection Using Eigen Values. *International Journal of Information and Electronics Engineering*, 5(1):40–45, 2015.

[16] F. N. Bezerra and E. Lima. Low Cost Soccer Video Summaries based on Visual Rhythm. In *8th ACM International Workshop on Multimedia Information Retrieval*, pages 71–78, Santa Barbara, CA, USA, 2006.

[17] J. Boreczky, A. Girgensohn, G. Golovchinsky, and S. Uchihashi. An Interactive Comic Book Presentation for Exploring Video. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 185–192. ACM, 2000.

[18] M. Bornert, F. Brémand, P. Doumalin, J.-C. Dupré, M. Fazzini, M. Grédiac, F. Hild, S. Mistou, J. Molimard, J.-J. Orteu, L. Robert, Y. Surrel, P. Vacher, and B. Wattrisse. Assessment of digital image correlation measurement errors: Methodology and results. *Experimental Mechanics*, 49(3):353–370, 2009.

[19] T. N. Bui and C. Jones. Finding Good Approximate Vertex and Edge Partitions is NP-hard. *Information Processing Letters*, 42(3):153–159, May 1992.

[20] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *11th European Conference on Computer Vision: Part IV*, pages 778–792, Berlin, Heidelberg, 2010. Springer-Verlag.

[21] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla. Evaluation of Low-Complexity Visual Feature Detectors and Descriptors. In *18th International Conference on Digital Signal Processing*, pages 1–7, July 2013.

[22] D. M. Chandler. Seven Challenges in Image Quality Assessment: Past, Present, and Future Research. *International Scholarly Research Notices Signal Processing*, pages 1–53, 2013.

[23] V. Chasanis, A. Likas, and N. Galatsanos. Video Rushes Summarization Using Spectral Clustering and Sequence Alignment. In *2nd ACM TRECVid Video Summarization Workshop*, pages 75–79, Vancouver, BC, Canada, 2008.

[24] M. Chatzigiorgaki and A. N. Skodras. Real-Time Keyframe Extraction Towards Video Content Identification. In *16th International Conference on Digital Signal Processing*, pages 934–939, Piscataway, NJ, USA, 2009. IEEE Press.

[25] C. Chen and J. A. Bloom. A Blind Reference-free Blockiness Measure. In *11th Pacific Rim Conference on Advances in Multimedia Information Processing: Part I*, pages 112–123, Berlin, Heidelberg, 2010. Springer-Verlag.

[26] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam. Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison. *IEEE Transactions on Broadcasting*, 57(2):165–182, June 2011.

[27] M. G. Christel, A. G. Hauptmann, E. G. Hauptmann, A. S. Warmack, and S. A. Crosby. Adjustable Filmstrips and Skims as Abstractions for a Digital Video Library. In *IEEE Advances in Digital Libraries Conference*, pages 98–104. IEEE Press, 1999.

[28] W.-S. Chu, Y. Song, and A. Jaimes. Video Co-summarization: Video Summarization by Visual Co-Occurrence. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2015.

[29] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[30] G. Ciocca and R. Schettini. An Innovative Algorithm for Key Frame Extraction in Video Summarization. *Journal of Real-Time Image Processing*, 1(1):69–88, 2006.

[31] M. V. M. Cirne and H. Pedrini. Video Summarization Method Based on Spectral Clustering. In *18th Iberoamerican Congress on Pattern Recognition*, volume 8259, pages 479–486, Havana, Cuba, 2013.

[32] M. V. M. Cirne and H. Pedrini. Summarization of Videos by Image Quality Assessment. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 8827 of *Lecture Notes in Computer Science*, pages 901–908. Springer International Publishing, 2014.

[33] F. Coldefy and P. Bouthemy. Unsupervised Soccer Video Abstraction based on Pitch, Dominant Color and Camera Motion Analysis. In *12th annual ACM International Conference on Multimedia*, pages 268–271, New York, NY, USA, 2004. ACM.

[34] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines: and other Kernel-Based Learning Methods*. Cambridge University Press, New York, NY, USA, 2000.

[35] T. O. Cunha, F. G. H. Souza, G. L. Pappa, and A. A. Araújo. VSRV: Video Summarization for Rushes Videos. In *XXV SIBGRAPI Conference on Graphics, Patterns and Images, Workshop de Teses e Dissertações*, pages 172–177, Ouro Preto, MG, Brasil, 2012.

[36] U. Damnjanovic, V. Fernandez, E. Izquierdo, and J. M. Martinez. Event Detection and Clustering for Surveillance Video Summarization. In *Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, pages 63–66, Washington, DC, USA, 2008. IEEE Computer Society.

[37] U. Damnjanovic, E. Izquierdo, and M. Grzegorzek. Shot Boundary Detection Using Spectral Clustering. In M. Domanski, R. Stasinski, and M. Bartkowiak, editors,

15th European Signal Processing Conference, pages 1779–1783, Poznan, Poland, Sept. 2007. PTETiS, Poznan.

[38] S. E. F. de Avila, A. P. B. Lopes, A. da Luz Jr., and A. de Albuquerque Araújo. VSUMM: A Mechanism Designed to Produce Static Video Summaries and a Novel Evaluation Method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[39] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edition, 2008.

[40] S. De Bruyne, J. De Cock, C. Poppe, C.-F. Hollemeersch, P. Lambert, and R. Van de Walle. Compressed-Domain Shot Boundary Detection for H.264/AVC Using Intra Partitioning Maps. In *Advances in Multimedia Modeling*, volume 6523 of *Lecture Notes in Computer Science*, pages 29–39. Springer Berlin Heidelberg, 2011.

[41] L. G. L. B. M. de Vasconcelos. Sumarização Automática em Melhores Momentos de Transmissões Televisivas de Futebol. Master's thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2011.

[42] L. Debing, C. Zhibo, M. Huadong, X. Feng, and G. Xiaodong. No-Reference Block Based Blur Detection. In *International Workshop on Quality of Multimedia Experience*, pages 75–80, July 2009.

[43] K. Demirtas, N. K. Cicekli, and I. Cicekli. Automatic Categorization and Summarization of Documentaries. *Journal of Information Science*, 36(6):671–689, Dec. 2010.

[44] L. Di Stefano, S. Mattoccia, and F. Tombari. ZNCC-based Template Matching Using Bounded Partial Correlation. *Pattern Recognition Letters*, 26(14):2129–2134, Oct. 2005.

[45] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic Soccer Video Analysis and Summarization. *IEEE Transactions on Image Processing*, 12(7):796–807, July 2003.

[46] E. Elhamifar, G. Sapiro, and R. Vidal. See All by Looking at a Few: Sparse Modeling for Finding Representative Objects. In *IEEE Computer Vision and Pattern Recognition*, pages 1600–1607, Los Alamitos, CA, USA, 2012.

[47] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, OR, USA, 1996.

[48] D. Farin, W. Effelsberg, and P. H. N. de With. Robust Clustering-Based Video-Summarization with Integration of Domain-Knowledge. In *IEEE International Conference on Multimedia and Expo*, pages 89–92. IEEE, 2002.

[49] A. Friedman. Framing Pictures: The Role of Knowledge in Automatized Encoding and Memory for Gist. *Journal of Experimental Psychology: General*, 108:316–355, 1979.

[50] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini. STIMO: STIll and MOving Video Storyboard For The Web Scenario. In *Multimedia Tools and Applications*, volume 46, pages 47–69, Hingham, MA, USA, 2010. Kluwer Academic Publishers.

[51] Y. Gong and X. Liu. Video Summarization and Retrieval Using Singular Value Decomposition. *Multimedia Systems*, 9(2):157–168, Aug. 2003.

[52] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[53] T. F. Gonzalez. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science*, 38:293–306, 1985.

[54] S. J. F. Guimarães, M. Couprie, A. d. A. Araújo, and N. J. Leite. Video Segmentation Based on 2D Image Analysis. *Pattern Recognition Letters*, 24(7):947–957, Apr. 2003.

[55] Y. Hadi, F. Essannouni, and R. O. H. Thami. Video Summarization by K-Medoid Clustering. In *ACM Symposium on Applied Computing*, pages 1400–1401, New York, NY, USA, 2006. ACM.

[56] L. Hagen and A. B. Kahng. New Spectral Methods for Ratio Cut Partitioning and Clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, Sept. 1992.

[57] A. Hanjalic and H. Zhang. An Integrated Scheme for Automated Video Abstraction Based on Unsupervised Cluster-Validity Analysis. *IEEE Transactions on Circuits Systems for Video Technology*, 9(8):1280–1289, 1999.

[58] R. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, Nov. 1973.

[59] L. He, E. Sanocki, A. Gupta, and J. Grudin. Auto-Summarization of Audio-Video Presentations. In *Seventh ACM International Conference on Multimedia (Part 1)*, pages 489–498. ACM, 1999.

[60] J. Heinly, E. Dunn, and J.-M. Frahm. Comparative Evaluation of Binary Features. In *12th European Conference on Computer Vision - Volume Part II*, pages 759–773, Berlin, Heidelberg, 2012. Springer-Verlag.

[61] F. Hild and S. Roux. Comparison of Local and Global Approaches to Digital Image Correlation. *Experimental Mechanics*, 52(9):1503–1519, 2012.

[62] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. Nguyen. Generalized Hessian Matrix and Second-Order Optimality Conditions for Problems with $C^{1,1}$ Data. *Applied Mathematics and Optimization*, 11(1):43–56, 1984.

[63] A. Hore and D. Ziou. Image Quality Metrics: PSNR vs. SSIM. In *20th International Conference on Pattern Recognition*, pages 2366–2369, Aug. 2010.

[64] M. B. Islam, K. Kundu, and A. Ahmed. Texture Feature Based Image Retrieval Algorithms. *International Journal of Engineering and Technical Research*, 2:170–173, Apr. 2014.

[65] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[66] N. Janwe and K. Bhoyar. Video Shot Boundary Detection Based on JND Color Histogram. In *IEEE Second International Conference on Image Information Processing*, pages 476–480, Dec. 2013.

[67] A. Jensen and A. La Cour-Harbo. *Ripples in Mathematics : The Discrete Wavelet Transform*. Springer, Berlin, 2001.

[68] T. Kailath. The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, Feb. 1967.

[69] J. Kavitha and P. A. J. Rani. Static and Multiresolution Feature Extraction for Video Summarization. *Procedia Computer Science*, 47(0):292–300, 2015.

[70] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 506–513, Washington, DC, USA, 2004. IEEE Computer Society.

[71] J.-G. Kim, H. S. Chang, K. Kang, M. Kim, J. Kim, and H.-M. Kim. Summarization of News Video and its Description for Content-Based Access. *International Journal of Imaging Systems and Technology*, 13(5):267–274, 2003.

[72] R. Kloss, S. Silva, W. Schwartz, M. Cirne, and H. Pedrini. Partial Least Squares Image Clustering. In *Conference on Graphics, Patterns and Images (XXVIII SIBGRAPI)*, pages 1–8, Salvador-BA, Brazil, 2015.

[73] A. Koz and A. A. Alatan. Oblivious Spatio-Temporal Watermarking of Digital Video by Exploiting the Human Visual System. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(3):326–337, Mar. 2008.

[74] I. Laptev. On Space-Time Interest Points. *International Journal of Computer Vision*, 64(2-3):107–123, Sept. 2005.

[75] A. L. Lavanya and R. Sreepada. A Generic Frame Work for Image Data Clustering Via Weighted Clustering Ensemble. *International Journal of Computer Science & Information Technologies*, 3:5429–5433, Nov. 2012.

[76] D.-D. Le and S. Satoh. National Institute of Informatics, Japan at TRECVID 2007: BBC Rushes Summarization. In *International Workshop on TRECVID Video Summarization*, pages 70–73, New York, NY, USA, 2007. ACM.

[77] K. Lee and M. Kolsch. Shot Boundary Detection with Graph Theory Using Keypoint Features and Color Histograms. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1177–1184, Jan. 2015.

[78] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering Important People and Objects for Egocentric Video Summarization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353, 2012.

[79] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *International Conference on Computer Vision*, pages 2548–2555, Washington, DC, USA, 2011. IEEE Computer Society.

[80] J. P. Lewis. Fast Normalized Cross-Correlation. *Vision Interface*, 10(1):120–123, 1995.

[81] B. Li, H. Pan, and I. Sezan. A General Framework for Sports Video Summarization with its Application to Soccer. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 169–172, 2003.

[82] Q. Li and Z. Wang. Reduced-Reference Image Quality Assessment Using Divisive Normalization-Based Image Representation. *IEEE Journal of Selected Topics in Signal Processing*, 3(2):202–211, Apr. 2009.

[83] R. Lienhart. Dynamic Video Summarization of Home Video. In *Storage and Retrieval for Media Databases*, pages 378–389, 2000.

[84] T. Lin and H. J. Zhang. Automatic Video Scene Extraction by Shot Grouping. *International Conference on Pattern Recognition*, 4:39–42, 2000.

[85] H. Liu, N. Klomp, and I. Heynderickx. A No-Reference Metric for Perceived Ringing Artifacts in Images. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(4):529–539, Apr. 2010.

[86] D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[87] Z. Lu and K. Grauman. Story-Driven Summarization for Egocentric Video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2721, Washington, DC, USA, 2013. IEEE Computer Society.

[88] U. Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416, Dec. 2007.

[89] K. M. Mahmoud, M. A. Ismail, and N. M. Ghanem. VSCAN: An Enhanced Video Summarization Using Density-Based Spatial Clustering. In *Lecture Notes in Computer Science*, volume 8156, pages 733–742. Springer, 2013.

[90] L. Z. Manor and P. Perona. Self-Tuning Spectral Clustering. In *Advances in Neural Information Processing Systems*, volume 17, pages 1601–1608. MIT Press, 2004.

[91] A. G. Money and H. Agius. Video Summarisation: A Conceptual Framework and Survey of the State of the Art. *Journal of Visual Communication and Image Representation*, 19(2):121–143, Feb. 2008.

[92] M. C. Morrone and D. C. Burr. Feature Detection in Human Vision: A Phase-Dependent Energy Model. *Proceedings of the Royal Society of London B: Biological Sciences*, 235(1280):221–245, Dec. 1988.

[93] P. Mundur, Y. Rao, and Y. Yesha. Keyframe-Based Video Summarization Using Delaunay Clustering. *International Journal on Digital Libraries*, 6:219–232, Apr. 2006.

[94] A. Nasiri Avanaki, S. Sodagari, and A. Diyanat. Reduced Reference Image Quality Assessment Metric Using Optimized Parameterized Wavelet Watermarking. In *9th International Conference on Signal Processing*, pages 868–871, Oct. 2008.

[95] A. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[96] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang. Video summarization and scene detection by graph modeling. *IEEE Transactions on. Circuits and Systems for Video Technology*, 15:296–305, 2005.

[97] NVIDIA CUDA C Programming Guide Version, 2015. `http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html`.

[98] J.-N. Ouellet and V. Randrianarisoa. To Watch or Not to Watch: Video Summarization with Explicit Duplicate Elimination. In *Canadian Conference on Computer and Robot Vision*, pages 340–346, Washington, DC, USA, 2011. IEEE Computer Society.

[99] P. Over, A. F. Smeaton, and G. Awad. The TRECVID 2008 BBC Rushes Summarization Evaluation. In *2nd ACM TRECVID Video Summarization Workshop*, pages 1–20, New York, NY, USA, 2008. ACM.

[100] G. Pal, D. Rudrapaul, S. Acharjee, R. Ray, S. Chakraborty, and N. Dey. Video Shot Boundary Detection: A Review. In S. C. Satapathy, A. Govardhan, K. S. Raju, and J. K. Mandal, editors, *Emerging ICT for Bridging the Future - Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2*, volume 338 of *Advances in Intelligent Systems and Computing*, pages 119–127. Springer International Publishing, 2015.

[101] B. Pan and Z. Wang. Recent Progress in Digital Image Correlation. In *Application of Imaging Techniques to Mechanics of Materials and Structures, Volume 4*, Conference Proceedings of the Society for Experimental Mechanics Series, pages 317–326. Springer New York, 2013.

[102] B. Pan, H. Xie, and Z. Wang. Equivalence of Digital Image Correlation Criteria for Pattern Matching. *Applied Optics*, 49(28):5501–5509, Oct. 2010.

[103] C.-M. Pan, Y.-Y. Chuang, and W. H. Hsu. NTU TRECVID-2007 Fast Rushes Summarization System. In *International Workshop on TRECVID Video Summarization*, pages 74–78, New York, NY, USA, 2007. ACM.

[104] U. Patel, P. Shah, and P. Panchal. Shot Detection Using Pixel wise Difference with Adaptive Threshold and Color Histogram Method in Compressed and Uncompressed Video. *International Journal of Computer Applications*, 64(4):38–44, Feb. 2013.

[105] J. Peng and Q. Xiaolin. Keyframe-Based Video Summary Using Visual Attention Clues. *IEEE MultiMedia*, 17(2):64–73, 2010.

[106] W.-T. Peng, W.-J. Huang, W.-T. Chu, C.-N. Chou, W.-Y. Chang, C.-H. Chang, and Y.-P. Hung. A User Experience Model for Home Video Summarization. In *15th International Multimedia Modeling Conference on Advances in Multimedia Modeling*, pages 484–495. Springer-Verlag, 2008.

[107] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg. Abstracting Digital Movies Automatically. *Journal of Visual Communication and Image Representation*, 7:345–353, 1996.

[108] D. Rao and L. Reddy. Contrast Weighted Perceptual Structural Similarity Index for Image Quality Assessment. In *Annual IEEE India Conference*, pages 1–4, Dec. 2009.

[109] A. Rehman and Z. Wang. Reduced-Reference SSIM Estimation. In *17th IEEE International Conference on Image Processing*, pages 289–292, Sept. 2010.

[110] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color Transfer Between Images. *IEEE Computer Graphics and Applications*, 21(5):34–41, Sept. 2001.

[111] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision*, Barcelona, Spain, 2011.

[112] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[113] G. Sanguinetti, J. Laidler, and N. D. Lawrence. Automatic Determination of the Number of Clusters Using Spectral Algorithms. In *IEEE Machine Learning for Signal Processing*, pages 28–30, 2005.

[114] C. Sharma and P. Sathish. Parallelizing Keyframe Extraction for Video Summarization. In *International Conference on Signal Processing And Communication Engineering Systems*, pages 245–249, Jan. 2015.

[115] H. R. Sheikh and A. C. Bovik. Image Information and Visual Quality. *IEEE Transactions on Image Processing*, 15(2):430–444, Feb. 2006.

[116] B. Shekar, K. Raghurama Holla, and M. Sharmila Kumari. Video Shot Detection Using Cumulative Colour Histogram. In *4th International Conference on Signal and Image Processing*, volume 222, pages 353–363. Springer India, 2012.

[117] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22, pages 888–905, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[118] N. C. Simões. Detecção de Algumas Transições Abruptas em Sequências de Imagens. Master's thesis, Universidade Estadual de Campinas, Campinas, SP, Brasil, 2004.

[119] R. Singh and N. Aggarwal. Novel Research in the Field of Shot Boundary Detection – A Survey. In *Advances in Intelligent Informatics*, volume 320 of *Advances in Intelligent Systems and Computing*, pages 457–469. Springer International Publishing, 2015.

[120] M. A. Smith. Video Skimming and Characterization Through the Combination of Image and Language Understanding. In *International Workshop on Content-Based Access of Image and Video Databases*, pages 61–70, Washington, DC, USA, 1998. IEEE Computer Society.

[121] R. Soundararajan and A. C. Bovik. RRED Indices: Reduced Reference Entropic Differencing for Image Quality Assessment. *IEEE Transactions on Image Processing*, 21(2):517–526, 2012.

[122] Y. Takahashi, N. Nitta, and N. Babaguchi. Video Summarization for Large Sports Video Archives. *IEEE International Conference on Multimedia and Expo*, pages 1170–1173, 2005.

[123] C. Taskiran, E. Delp, and T. Reed. *Digital Image Sequence Processing, Compression, and Analysis*. CRC Press Computer Engineering Series. CRC Press, 2005.

[124] B. T. Truong and S. Venkatesh. Video Abstraction: A Systematic Review and Classification. *ACM Transactions on Multimedia Computing, Communications and Applications*, 3(1), Feb. 2007.

[125] S. Uchihashi, J. Foote, A. Girgensohn, and J. S. Boreczky. Video Manga: Generating Semantically Meaningful Video Summaries. In *7th ACM International Conference on Multimedia*, pages 383–392, Orlando, FL, USA, 1999.

[126] M. Unser. Sum and Difference Histograms for Texture Classification. *IEEE Trans. Pattern Analysis Machine Intelligence*, 8(1):118–125, Jan. 1986.

[127] D. Verma and M. Meila. A Comparison of Spectral Clustering Algorithms. Technical report, University of Washington, 2003.

[128] D. Wagner and F. Wagner. Between Min Cut and Graph Bisection. In *18th International Symposium on Mathematical Foundations of Computer Science*, pages 744–750, London, UK, 1993. Springer-Verlag.

[129] B. Walter, S. N. Pattanaik, and D. P. Greenberg. Using Perceptual Texture Masking for Efficient Image Synthesis. *Computer Graphics Forum*, pages 393–400, 2002.

[130] J. Wang, Y. Wang, and Z. Zhang. Visual Saliency Based Aerial Video Summarization by Online Scene Classification. In *Sixth International Conference on Image and Graphics*, pages 777–782, Washington, DC, USA, 2011. IEEE Computer Society.

[131] Z. Wang and A. C. Bovik. A Universal Image Quality Index. *IEEE Signal Processing Letters*, 9(3):81–84, Mar. 2002.

[132] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[133] Z. Wang and E. P. Simoncelli. Reduced-Reference Image Quality Assessment Using a Wavelet-Domain Natural Image Statistic Model. In *Proceedings of SPIE, X Conference on Human Vision and Electronic Imaging*, volume 5666, pages 149–159, San Jose, CA, 1 2005.

[134] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale Structural Similarity for Image Quality Assessment. In *37th Asilomar Conf on Signals, Systems and Computers*, volume 2, pages 1398–1402. IEEE Computer Society, 11 2003.

[135] S. Wold, K. Esbensen, and P. Geladi. Principal Component Analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1–3):37–52, 1987. Multivariate Statistical Workshop for Geologists and Geochemists.

[136] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh. Gaze-Enabled Egocentric Video Summarization via Constrained Submodular Maximization. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[137] I. Yahiaoui, B. Mérialdo, and B. Huet. Generating Summaries Of Multi-Episode Video. In *IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, Aug. 2001.

[138] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating Bag-of-visual-words Representations in Scene Classification. In *International Workshop on Workshop on Multimedia Information Retrieval*, pages 197–206, New York, NY, USA, 2007. ACM.

[139] K. Yang and H. Jiang. Optimized-SSIM Based Quantization in Optical Remote Sensing Image Compression. In *International Conference on Image and Graphics*, pages 117–122, Los Alamitos, CA, USA, 2011. IEEE Computer Society.

[140] H. Yi, Z. Pengzhou, and W. Yanfeng. Adaptive Threshold Based Video Shot Boundary Detection Framework. In *International Conference on Image Analysis and Signal Processing*, pages 1–5, Nov. 2012.

[141] B. Yu, W.-Y. Ma, K. Nahrstedt, and H.-J. Zhang. Video Summarization Based on User Log Enhanced Link Analysis. In *Eleventh ACM International Conference on Multimedia*, pages 382–391. ACM, 2003.

[142] H. M. Zawbaa, N. El-Bendary, A. E. Hassanien, and A. Abraham. SVM-Based Soccer Video Summarization System. In *NaBIC*, pages 7–11, 2011.

[143] H. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic Partitioning of Full-motion Video. *Multimedia Systems*, 1(1):10–28, Jan. 1993.

[144] H. Zhang and S. W. Smoliar. Developing Power Tools for Video Indexing and Retrieval. In *Storage and Retrieval for Image and Video Databases*, pages 140–149, 1994.

[145] L. Zhang, L. Zhang, X. Mou, and D. Zhang. FSIM: A Feature Similarity Index for Image Quality Assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, Aug. 2011.

[146] S.-H. Zhong, Y. Liu, Y. Liu, and F.-L. Chung. A Semantic No-Reference Image Sharpness Metric Based on Top-Down and Bottom-Up Saliency Map Modeling. In *17 IEEE International Conference on Image Processing*, pages 1553–1556, Sept. 2010.

[147] H. Zhou, A. H. Sadka, M. R. Swash, J. Azizi, and U. A. Sadiq. Feature Extraction and Clustering for Dynamic Video Summarisation. *Neurocomputing*, 73:1718–1729, June 2010.

[148] X. Zhu, J. Fan, A. K. Elmagarmid, and X. Wu. Hierarchical Video Content Description and Summarization Using Unified Semantic and Visual Similarity. *Multimedia Systems*, 9(1):31–53, July 2003.

Appendix A

List of Videos from Open Video Project

Table A.1 shows a list of the 50 videos from the OVP database used in the tests of each method described in this thesis.

Video Title	**#Frames**	**Duration**
The Great Web of Water, segment 01	3,279	1:50
The Great Web of Water, segment 02	2,118	1:11
The Great Web of Water, segment 07	1,745	0:59
A New Horizon, segment 01	1,806	1:01
A New Horizon, segment 02	1,797	1:00
A New Horizon, segment 03	6,249	3:29
A New Horizon, segment 04	3,192	1:47
A New Horizon, segment 05	3,561	1:59
A New Horizon, segment 06	1,944	1:05
A New Horizon, segment 08	1,815	1:01
New Horizon, segment 10	2,517	1:24
Take Pride in America, segment 01	2,691	1:30
Take Pride in America, segment 03	3,261	1:49
Digital Jewelry: Wearable Technology for Every Day Life	4,204	3:00
HCIL Symposium 2002 - Introduction, segment 01	2,336	1:18
Senses And Sensitivity, Introduct. to Lecture 1 presenter	4,221	2:20
Senses And Sensitivity, Introduct. to Lecture 2	3,411	1:53
Senses And Sensitivity, Introduct. to Lecture 3 presenter	4,566	2:32
Senses And Sensitivity, Introduct. to Lecture 4 presenter	5,249	2:55
Exotic Terrane, segment 01	2,940	1:38
Exotic Terrane, segment 02	2,776	1:32
Exotic Terrane, segment 03	2,676	1:29
Exotic Terrane, segment 04	4,797	2:40
Exotic Terrane, segment 06	2,425	1:21
Exotic Terrane, segment 08	2,428	1:21
America's New Frontier, segment 01	3,591	1:59
America's New Frontier, segment 03	2,166	1:12

Continued on next page

Table A.1 – continued from previous page		
Video Title	**#Frames**	**Duration**
America's New Frontier, segment 04	3,705	2:03
America's New Frontier, segment 07	3,615	2:00
America's New Frontier, segment 10	4,830	2:41
The Future of Energy Gases, segment 03	2,934	1:37
Future of Energy Gases, segment 05	3,615	2:00
The Future of Energy Gases, segment 09	1,884	1:02
The Future of Energy Gases, segment 12	2,886	1:36
Oceanfloor Legacy, segment 01	1,740	0:58
Oceanfloor Legacy, segment 02	2,325	1:17
Oceanfloor Legacy, segment 04	3,450	1:55
Oceanfloor Legacy, segment 08	3,186	1:46
Oceanfloor Legacy, segment 09	2,106	1:10
The Voyage of the Lee, segment 05	2,094	1:09
The Voyage of the Lee, segment 15	2,277	1:15
The Voyage of the Lee, segment 16	2,619	1:27
Hurricane Force - A Coastal Perspective, segment 03	2,310	1:17
Hurricane Force - A Coastal Perspective, segment 04	5,310	2:57
Drift Ice as a Geologic Agent, segment 03	2,742	1:31
Drift Ice as a Geologic Agent, segment 05	2,187	1:12
Drift Ice as a Geologic Agent, segment 06	2,425	1:30
Drift Ice as a Geologic Agent, segment 07	1,950	1:05
Drift Ice as a Geologic Agent, segment 08	3,618	2:00
Drift Ice as a Geologic Agent, segment 10	1,407	0:46

Table A.1: Average precision, recall and F-measures of the summaries produced by each method for the entire database.