A New Method for Static Video Summarization Using Visual Words and Video Temporal Segmentation

Edward Jorge Yuri Cayllahua Cahuina
Federal University of Ouro Preto

Dissertation submitted to the
Department of Computer Science (DECOM)
Institute of Biological and Exact Sciences (ICEB)
Federal University of Ouro Preto
for the Degree of Master in Computer Science
Minas Gerais, Brazil
Cahuina, Edward Jorge Yuri Cayllahua.


88f.: il.; grafs.; tabs.

Orientador: Prof. Guillermo Cámara Chávez.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Departamento de Computação. Programa de Pós-graduação em Ciência da Computação.
Área de concentração: Recuperação e Tratamento da Informação.


CDU:004.424.5:621.397.4

Catalogação: sisbin@sisbin.ufop.br
Ata da Defesa Pública de Dissertação de Mestrado

Aos 22 dias do mês de fevereiro de 2013, às 10 horas na Sala de Seminários do DECOM no Instituto de Ciências Exatas e Biológicas (ICEB), reuniram-se os membros da banca examinadora composta pelos professores: Prof. Dr. Guillermo Câmara Chávez (presidente e orientador), Prof. Dr. David Menotti Gomes, Prof. Dr. Arnaldo de Albuquerque Araújo e Prof. Dr. Silvio Jamil Ferzoli Guimarães, aprovada pelo Colegiado do Programa de Pós-Graduação em Ciência da Computação, a fim de arguirem o mestrando Edward Jorge Yuri Cayllahua Cashuina, com o título “Static Video Summarization Based on Local Descriptors and Temporal Segmentation". Aberta a sessão pelo presidente, coube ao candidato, na forma regimental, expor o tema de sua dissertação, dentro do tempo regulamentar, sendo em seguida questionado pelos membros da banca examinadora, tendo dado as explicações que foram necessárias.

Recomendações da Banca:
( ) Aprovada sem recomendações
( ) Reprovada
( y) Aprovada com recomendações: ________________________________

Banca Examinadora:

Prof. Dr. Guillermo Câmara Chávez

Prof. Dr. David Menotti Gomes

Prof. Dr. Arnaldo de Albuquerque Araújo

Prof. Dr. Silvio Jamil Ferzoli Guimarães

Prof. Dr. Haroldo Gambini Santos

Coordenador do Programa de Pós-Graduação em Ciência da Computação DECOM/ICEB/UFO

Ouro Preto, 22 de fevereiro de 2013.
A New Method for Static Video Summarization Using Visual Words and Video Temporal Segmentation

Abstract

During the last years, a continuous demand and creation of digital video information have occurred. The creation of digital video has caused an exponential growth of digital video content. To increase the usability of such large volume of videos, a lot of research has been made. Video summarization, in particular, has been proposed to rapidly browse large video collections. It has also been used to efficiently index and access video content. To summarize any type of video, researchers have relied on visual features contained in frames. In order to extract these features, different techniques have used local or global descriptors. Nonetheless, no extensive evaluation have been made about the usefulness of both types of descriptors in video summarization.

One important contribution of this dissertation is to propose a method for semantic video summarization that can produce meaningful and informative video summaries. In this dissertation, we perform a wide evaluation using over 100 videos; in order to achieve a stronger position about the performance of local descriptors in semantic video summarization. According to our experiments, our proposed method using local descriptors and temporal video segmentation produce better summaries than other methods that do not. We also acknowledge a marginal importance of color information when using local descriptors to produce video summaries.
Durante os últimos anos, uma demanda contínua de informações de vídeo digital ter ocorrido. A criação de vídeo digital tem provocado um crescimento exponencial de conteúdo de vídeo digital. Para aumentar a usabilidade de grande volume de vídeos, muita pesquisa tem sido feita. A Sumarização Automática de Vídeos, em particular, tem sido proposto para explorar rapidamente grandes coleções de vídeo. Os resumos de vídeos tem sido utilizado de forma eficiente para indexar e conteúdos de vídeo de acesso. Para resumir qualquer tipo de vídeo, os pesquisadores tem usado as características visuais contidas nos quadros do vídeo. A fim de extrair essas características, diferentes técnicas têm utilizado descritores locais ou globais. No entanto, nenhuma avaliação extensa têm sido feita sobre a utilidade de ambos tipos de descritores na sumarização automática de vídeos.

Neste trabalho, realizamos uma ampla avaliação, a fim de alcançar uma posição mais forte sobre o desempenho de descritores locais na sumarização automática de vídeos. De acordo com nossos experimentos, nosso modelo proposto utilizando descritores locais e segmentação temporal de vídeos elabora resumos melhores do que os outros modelos que não. Nós também reconhecemos a importância marginal de informação de cor usada pelos descritores locais para produzir resumos de vídeo. Uma contribuição importante deste trabalho é propor um modelo simples para sumarização de vídeo que pode produzir resumos de vídeo significativos e informativos.
Acknowledgements

First and above all, I praise God, for providing me this opportunity and granting me
the capability to proceed successfully. This dissertation appears in its current form due
to the support and guidance of several people. I would therefore like to offer my sincere
thanks to all of them.

I would like to express my gratitude to my advisor and friend, Professor Guillermo
Camara Chavez, whose guidance and patience has been of great help and made possible
this dissertation to finish.

I would not have contemplated this road if not for my parents, Jorge and Benilda.
To my parents, thank you for your continuous support which encouraged me to study
abroad.

To my dear friend, Rensso Mora Colque, whose friendship made my stay at Brazil a
good experience.

And last, but not least, to Karla Otiniano Rodriguez, my girlfriend, my companion
and my dearest friend. This dissertation was produced thanks to her constant support
and companion.
# Conteúdo

<table>
<thead>
<tr>
<th>Capítulo</th>
<th>Título</th>
<th>Página</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Introduction</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>State of Art</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Methods Based on Image Descriptors</td>
<td>14</td>
</tr>
<tr>
<td>2.4</td>
<td>Methods Based on Text (Video Annotation)</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>Methods Based on Audio</td>
<td>16</td>
</tr>
<tr>
<td>2.6</td>
<td>Final Considerations</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>Theoretical Fundamentals</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Digital Video</td>
<td>19</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Terminology</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>Visual Effects in Videos</td>
<td>21</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Cut</td>
<td>21</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Fades and Dissolves</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>Image Descriptors</td>
<td>23</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Color Histogram</td>
<td>23</td>
</tr>
</tbody>
</table>

# Lista de Figuras

<table>
<thead>
<tr>
<th>Título</th>
<th>Página</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>xi</td>
</tr>
</tbody>
</table>

# Lista de Tabelas

<table>
<thead>
<tr>
<th>Título</th>
<th>Página</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
3.3.2 Projection Histograms ........................................... 25
3.3.3 Scale-Invariant Feature Transform (SIFT) ....................... 25
3.3.4 A SIFT Descriptor with Color Invariant Characteristics (CSIFT) ........................................... 30
3.3.5 HUESIFT ........................................................... 31
3.3.6 SURF: Speeded-Up Robust Features ................................. 31
3.3.7 Space-Time Interest Points (STIP) ............................... 32
3.3.8 Histograms of Oriented Gradients (HoG) ......................... 35
3.4 Classification ........................................................... 37
3.4.1 K-means .............................................................. 37
3.4.2 X-means: Extending K-means with Efficient Estimation of the Number of Clusters ........................................... 38
3.5 The Bag-of-Words model ............................................. 40
3.6 Final considerations .................................................... 40

4 Proposed Method ....................................................... 43
4.1 Proposed Method for Summarizing Videos ......................... 43
4.1.1 Temporal Video Segmentation .................................... 44
4.1.2 Detection of Representative Frames ............................... 46
4.1.3 Bag of Words (BoW) ................................................. 47
4.1.4 Histogram of Visual Words ............................... 48
4.1.5 Visual Word Vectors Clustering .................................... 48
4.1.6 Filter Results .......................................................... 49
4.2 Final Considerations .................................................. 49

5 Experimental Results .................................................. 53
5.1 Data Set ............................................................... 53
5.2 Experimental setup ..................................................... 56
5.3 Video Summary Evaluation ........................................... 57
5.4 Experiments ............................................................ 59
5.4.1 Results for the Open Video Database ........................................... 59
5.4.2 Results for the Youtube Database ........................................... 61
5.5 Final Results Analysis ................................................. 62
5.6 Final Considerations .................................................. 63

6 Conclusions and Future Work ...................................... 67
6.1 Future Work ............................................................ 68
# Lista de Figuras

1.1 A general approach for video summarization ........................................... 4

1.2 Illustration of a video sequence with shots and transitions. Shots are $S_1, S_2, S_3$ and $S_4$, TP is the transition period. ................................. 6

3.1 Anatomy of a video .......................................................... 20

3.2 A fade-in transition ......................................................... 22

3.3 A fade-out transition ......................................................... 23

3.4 A dissolve transition ........................................................ 23

3.5 Difference of Gaussians ..................................................... 27

3.6 Locating maxima/minima in DoG images ................................... 28

3.7 The Bag-of-Words model .................................................... 41

4.1 Method for Summarizing Videos ........................................... 44

4.2 A dissimilarity vector computed from video HCIL Symposium 2002 - Introduction, segment 01 .......................................................... 45

4.3 A refined dissimilarity vector computed from video HCIL Symposium 2002 - Introduction, segment 01 ......................................................... 50

4.4 Dissolve effect in video HCIL Symposium 2002 - Introduction, segment 01 between frames 2139 - 2162 ......................................................... 50

4.5 Variance vector for video HCIL Symposium 2002 - Introduction, segment 01. The circles are the detected dissolve effects ........................................ 51
5.1 User Summary Number 1 for video *Drift Ice as a Geologic Agent, segment 08* ......................................................... 63
5.2 User Summary Number 2 for video *Drift Ice as a Geologic Agent, segment 08* ......................................................... 64
5.3 User Summary Number 3 for video *Drift Ice as a Geologic Agent, segment 08* ......................................................... 64
5.4 User Summary Number 4 for video *Drift Ice as a Geologic Agent, segment 08* ......................................................... 64
5.5 User Summary Number 5 for video *Drift Ice as a Geologic Agent, segment 08* ......................................................... 65
5.6 Video Summary using the VSUMM1 approach for video *Drift Ice as a Geologic Agent, segment 08* ................................. 65
5.7 Video Summary using SURF descriptor for video *Drift Ice as a Geologic Agent, segment 08* ......................................................... 66
Lista de Tabelas

5.1 Videos used for experiments ............................... 54
5.2 (Open Video Data Set) Experiments using temporal segmentation . . . . 60
5.3 (Open Video Data Set) Experiments without temporal segmentation . . . 61
5.4 (Youtube Data Set) Experiments using temporal segmentation . . . . 62
5.5 (Youtube Data Set) Experiments without temporal segmentation . . . . 62
Capítulo 1

Introduction

1.1 Introduction

The amount of multimedia information such as text, audio, still images, animation, and video is growing every day. The accumulated volume of this information can become a large collection of data. A video is a perfect example of a multimedia resource that is in continuous generation. It would be an arduous work if a human tries to process such a large volume of data and even, at a certain scale, it would be impossible.

As a consequence of video information growth, an enormous quantity of video is uploaded to the Internet. TV video information is generated every day and security cameras generate hours of video. Furthermore, a video requires the user to watch the whole content to retrieve its information. This means that the user is forced to exclusively watch the video and is unable to do anything else during this time.

Due to this increased use of video and the human effort taken to process it, new technologies need to be researched in order to manage effectively and efficiently such a quantity of information. Video summarization has recently been of interest for many researchers due to its importance in several applications such as information browsing and retrieval (Xiong, Zhou, Tian, Rui and TS 2006, Valdes and Martinez 2011). A video summary is a short version of an entire video sequence and aims to give to a user a synthetic and useful visual abstract of a video sequence.

The final video summary must be a synopsis that can easily be interpreted by the user. This means that the summary must be presented in a user-friendly manner. The most important goal is to provide users with a concise video representation so that the user...
have a quick idea about the content of the video (Furini, Geraci, Montangero, Pellegrini, a1, a2 and a3 2010). This will help him/her decide whether to watch or not the whole video. Generally speaking, the task of video summarization has been approached by using different methods to cluster the video content and therefore, detect the redundancy of the video content in order to summarize it (Gao, Wang, Yong and Gu 2009). In Figure 1.1, a generic approach for video summarization is presented. The general steps involved are: a video segmentation, then a feature extraction process is performed, afterwards a redundancy detection based on the features is applied and finally the video summary is generated.

![Figure 1.1: A general approach for video summarization](image)

According to Truong and Venkatesh (2007) and Money and Agius (2008), the video summary can be represented into two fashions: a static video summary (storyboard) and a dynamic video skimming. Dynamic video skimming, also known as moving storyboard, consists in selecting the most relevant small dynamic portions (video skims) of audio and video in order to generate the video summary. The result is another video of shorter length. On the other hand, static video summary, or storyboard, is interested in selecting the most relevant frames (keyframes) of a video sequence and generate the correspondent video summary, which will be a collection of still images (Gao, Wang, Yong and Gu 2009).

The main advantage of dynamic video skimming is that the final summary can contain audio and video making it more entertaining due to more expressiveness and they are
usually intended for final users; lately, this type of summary is in high demand (Xiang-Wei, Zhang, Zhao and Zhu 2009). Meanwhile, a static summary is more appropriate for indexing, browsing and retrieval. Furthermore, static video summaries have one important advantage, they allow the user to quickly have a general overview of the video content. On the other hand, in a dynamic video summary, the user is still required to watch a small version of the video in order to understand its content.

Both, static and dynamic video summaries can be computed independently of the genre of the video. However, some methods base their analysis on the type of video they are trying to summarize. For example, sports videos may be better summarized by the most important events like goals, fouls, etc., most of these events are based on motion information. Nevertheless, in a generic video, motion may not represent an important information because it may belong to a not relevant and isolated event.

In order to summarize a generic video, most of the methods (de Avila, ao Lopes, da Luz and de Albuquerque Araújo 2011, Papadopoulos, Chatzichristofis and Papamarkos 2011, Yuan, Lu, Wu, Huang and Yu 2011) have heavily relied on visual features computed from video frames. Visual features can be used to describe the global or local characteristics of an image. Many methods have based their analysis on global or local image descriptors. However, there has not been a wide evaluation about the performance of these two types of descriptors applied to video summarization. In (Kogler, del Fabro, Lux, Schoeffmann and Boeszoermenyi 2009), an evaluation was performed but their data set was limited to only 4 small videos, and therefore it is not possible to achieve strong conclusions about the performance of local descriptors in video summarization.

Another important consideration in video summarization is temporal video segmentation. This task is usually performed by detecting transitions between shots and is often applied as the first step in video summarization. A shot is defined as an image sequence that presents continuous action which is captured from a single operation of a single camera. Shots are joined together in the editing stage of video production to form the final video, using different transitions. There are two different types of transitions that can occur between shots: abrupt (discontinuous) shot transitions, also referred as cuts; or gradual (continuous) shot transitions, which include video editing special effects (fade-in, fade-out, dissolving, etc). These transitions can be defined as:

- cut: an instantaneous change of visual content from one shot to another.
- fade-in: a shot gradually appears from a constant image.
• fade-out: a shot gradually disappears from a constant image.
• dissolve: the current shot fades out while the next shot fades in.

In Figure 1.2, a visual explanation of the different transitions can be seen.

![Diagram of video sequence with shots and transitions]

**Figure 1.2:** Illustration of a video sequence with shots and transitions. Shots are $S_1, S_2, S_3$ and $S_4$. TP is the transition period.

A successful video temporal segmentation can lead into a better shot identification. Shots can be considered as the smallest indexing unit where no changes in scene content can be perceived and higher level concepts are often constructed by combining and analyzing the inter and intra shot relationships (Camara Chavez, Precioso, Cord, Phillip Foliguet and de A. Araujo 2007). Therefore, we must outline the importance of an effective video temporal segmentation.

However, in video summarization, most of the methods (de Avila, ao Lopes, da Luz and de Albuquerque Araújo 2011, Guan, Wang, Yu, Mei, He and Feng 2012, Furini, Geraci, Montangero, Pellegrini, a1, a2 and a3 2010) perform a simple video segmentation. This means that no visual effects, such as dissolves, are considered. Their approach is usually simple, as they only try to identify the cuts transitions in order to detect the video shots. Temporal video segmentation applied to video summarization has not been exploded by most of the researchers. And consequently, its importance has not been evaluated.
1.2 Motivation

Researchers have acknowledged the importance of effectively and efficiently manage and browse large video collections. That is the reason why video summarization has been a popular topic of research during the last years. In order to provide meaningful and relevant video summaries, different methods have been proposed. Some methods propose a semantic analysis with the intention of producing more informative summaries. These methods have relied on object detection techniques in order to extract visual information to perform a semantic analysis. To do this, local descriptors have been widely used to describe the visual information contained in frames when summarizing videos. Despite its extensive use, there has not been an ample evaluation about the performance of local descriptors in video summarization.

This dissertation is mainly motivated by the necessity to realize an evaluation about the performance of global and local descriptors applied to video summarization using a large data set of over 100 videos. And also, to evaluate the importance of temporal video segmentation applied to video summarization. In order to elaborate a method that is capable of handling the video information and extract the most relevant frames contained in the video, a method for video summarization based on semantic information and temporal segmentation is proposed. Thus, providing the user a more informative previous knowledge about the video content.

1.3 Aims and Objectives

In this section, the main and specific objectives pursued in this dissertation are presented.

1.3.1 General Objective

Propose a method for static video summarization using semantic information applied to generic videos.
1.3.2 Specific objectives

1. Carry out a wide evaluation about the performance of local descriptors applied to video summarization.
2. Evaluate the importance of temporal video segmentation applied to video summarization.
3. Analyze the robustness of local descriptors compared to global descriptors.
4. Study the relevance of color information used by some local descriptors applied to video summarization.
5. Perform experiments using large datasets with all types of videos.

1.4 Contributions

During the last years, several approaches for video summarization have been proposed. Most of these approaches base their analysis on local or global visual features computed by descriptors. Nonetheless, there has not been an evaluation about what type of descriptor is better for video summarization. The main contribution of this dissertation is to present a method for static video summarization using semantic information and video temporal segmentation. We also perform a wide evaluation in order to achieve a stronger position about the performance of local descriptors in a semantic video summarization. Furthermore, we evaluate the robustness of local descriptors compared to global descriptors.

Additionally, another important thing to outline is that some local descriptors use color information and others do not. So far there has not been any evaluation on wether local descriptors using color information give more meaningful summaries compared to other local descriptors. Therefore, we also inspect if color information can help local descriptors to produce better video summaries.

One essential operation in video summarization is temporal segmentation. Temporal segmentation is applied in the early stages of video summarization in order to detect the shots of the video for later analysis. Most of the approaches use a simple temporal segmentation where only cuts transitions are detected. We propose a more elaborated temporal segmentation to detect other types of transition (dissolves). Furthermore, we
evaluate how local descriptors are affected by temporal segmentation and its importance in giving better video summaries.

Another important contribution of this dissertation is to propose a simple method for semantic video summarization that can produce meaningful and informative video summaries.

1.5 Thesis Outline

This thesis is organized as follows. In Chapter 2, we present the state-of-art of video summarization. Later, in Chapter 3, we present some basic definitions that will hold within this document and we will also review some important techniques used in video summarization. Next, in Chapter 4, the proposed method for video summarization is explained. Then, in Chapter 5, we show evaluation criteria and we also analyse the experimental results obtained in this dissertation. Finally, in Chapter 6, the reached conclusions and the future work are presented.
Capítulo 2

State of Art

Video summarization has been an active topic of investigation. Several methods have been created in order to synthesize large hours of video and overcome this problem. In this chapter, we present a review of different approaches that have been proposed for video summarization.

2.1 Introduction

The process of video summarization extracts the most relevant frames or segments of a video and produce a final sequence of images or portions of video. The final result should be easy to understand by the final user. Video summarization has been a popular topic of investigation and has attracted the attention of many researchers, making it a research topic that has been investigated for several years. Nonetheless, video summarization still poses a challenging problem. Video abstraction is still largely in the research phase (Truong and Venkatesh 2007).

Different approaches for video summarization have been developed so far. We can classify video summarization by the final presentation of the video summary. We have two types of methods: static and dynamic. Static methods produce the final summary based on selected keyframes. Therefore, the final result is a slideshow of these keyframes. The main advantage of this type of summary is that the user can access the information quickly, they can easily have a global review of the whole video content. The main disadvantage is that no audio, text or any other information is shown, just the most important frames of the video. On the other hand, dynamic methods select the portions of video that are considered to be relevant. Then, these portions are joined together.
The result is a video that is smaller than the original and can contain audio information, it is a more comprehensible summary but the main disadvantage is that the user has to watch the whole summary video in order to understand the video content.

A different classification is proposed by Money and Agius (2008), they propose to classify video summarization methods in three categories: internal, external, and hybrid. The most popular approach for video summarization is the internal. It is called internal because the whole analysis is applied to the video stream only, so the methods only use the data that is self contained in the video. External methods use the information that is not necessarily contained in the video, such as manual annotation, labeling, etc. (Miyauchi, Babaguchi and Kitahashi 2003, Dagtas and Abdel-Mottaleb n.d.). The hybrid methods use a combination of both internal and external information.

Internal methods have the main advantage that the whole method can be applied automatically using only the information contained in the video. Truong and Venkatesh (2007) present the different features used in the these approaches, the most popular features used are: visual, text, audio, visual dynamics, camera motion, mid-level semantics. Most of these features can only be applied to specific types of video. For example, we can easily extract visual, text and audio features in a movie or a sitcom, but all these features are not always present in the video captured by a cellphone or a security video.

In order to automatically summarize a video, an approach has to first extract the different features present in a video. Then, based on some similarity, it distinguishes which parts of the video can be considered as relevant and then extract these parts to build the summary. Therefore, the first step is to extract the features for posterior analysis. We will now present some of the existing methods in the literature. Since our work is more related to visual features, special attention to the methods that use visual features will be taken.

2.2 Methods Based on Image Descriptors

Image descriptors are probably one of the most popular resources used in computer vision. A frame is an image synchronized in time. Applying an image descriptor on a frame will give us descriptions about the visual features contained in it. They describe the basic characteristics of the image, such as the texture, shape, color, motion, etc. Color histograms are one of the most important descriptors used in video summarization.
Color histograms have often been used to measure the similarity between two frames, which is useful when the method’s goal is to summarize the video based on redundancy elimination. Zhuang, Rui, Huang and Mehrotra (1998) use a color histogram as the main descriptor. The idea is to segment the video in shots and form groups using an unsupervised clustering algorithm. Every frame will belong to a certain cluster based on its color histogram. Then, the closest frame to each centroid is marked as a keyframe and extracted to build the storyboard. Recent researches still use the simplicity and power of color histograms, such as the methods proposed by Furini, Geraci, Montangero, Pellegrini, a1, a2 and a3 (2010) and de Avila, ao Lopes, da Luz and de Albuquerque Araújo (2011). They basically share the same general structure of histogram extraction and later perform an unsupervised clustering algorithm in order to produce the video summary.

Color histograms are usually vectors of high dimensionality. In order to overcome this problem, several methods have proposed to apply mathematical procedures on these feature vectors in order to reduce its dimensionality. Gong and Liu (2003) propose to use the singular value decomposition (SVD), later Mundur, Rao and Yesha (2006) and Wan and Qin (2010) use the principal component analysis (PCA). Also, in (Cahuina, Chavez and Menotti 2012), a static video summarization approach is presented using both color histograms and dimension reduction using PCA. Thus far, no comparisons have been performed between the two approaches. Furthermore, there has been no evaluation about the cost-benefit between the results obtained and the computational cost implied in performing these mathematical procedures.

In (Papadopoulos, Chatzichristofis and Papamarkos 2011), not only color descriptors but other visual descriptors are used. They propose to use the Compact Composite Descriptors (CCDs), which consist of four descriptors: the Color and Edge Directivity Descriptor (CEDD) and the the Fuzzy Color and Texture Histogram (FCTH) proposed by Zagoris, Ergina and Papamarkos (2010), the Brightness and Texture Directionality Histogram (BTDH) descriptor (Chatzichristofis and Boutalis 2010) and the Spatial Color Distribution Descriptor (SpCD) (Chatzichristofis, outalis and Lux 2010). They state that their method gives satisfactory results compared to four other methods, unfortunately they only use five videos to make the comparison.
2.3 Methods Based on Mid-level Semantics

A more informative summary can be obtained if the method considers the semantic meaning implied in the video. Semantic comprehension is *per se* an active topic of research in computer vision and is still in progress, so far no method has perfectly performed semantic comprehension on images. Videos are composed by consecutive images, and these images contain all types of objects. To delimitate the problem, some methods have based their analysis according to the interest of the final user and the input video that is going to be summarized. Miura, Hamada, Ide, Sakai and Tanaka (2003) try to summarize cooking videos performing a face detection procedure, the segments where faces are detected will not be considered in the final abstract since they have no cooking-related information. In contrast, (Peker, Otsuka, Divakaran, Peker, Otsuka and Divakaran 2006, Lee and Kim 2004, Peng, Chu, Chang, Chou, Huang, Chang and Hung 2011) perform face recognition in order to mark these segments as relevant ones, which is very important when summarizing movies, sitcoms or security videos. These methods are mainly interested in object recognition. In (zhiqiang Tian, Xue, Lan, li and Zheng 2011), a static video summarization based on object recognition is proposed. The idea is to eliminate redundancy of information from the temporal and spatial domain and also from the content domain by doing object recognition. The shot boundaries are detected and video objects are extracted using a 3D graph-based algorithm. Then, a K-means clustering algorithm is applied to detect the key objects.

In order to summarize sports videos, such as soccer, basketball, baseball, etc. The methods are more concerned about event detection. In (Fendri, Ben-Abdallah and Hamadou 2010), the method detects specific events (attacks, goals, etc.) and then produces a summary based on the event the user is more interested. Kim, Lee, Jung, Kim, Kim and Kim (2005) propose a method that not only performs event detection but also tracks the score. The goal is to give more importance to the previous moments before a ball goes into the basket, since they consider that finishing plays should be given more importance.

To summarize general videos taking into account the semantic information, the methods have relied on object detection. Yuan, Lu, Wu, Huang and Yu (2011) base their method on concept preservation. They use the Bag of Words (BoW) model. They first segment the video into samples/shots, then for each shot the SIFT descriptor (Scale-invariant feature transform) (Lowe 1999) is used to extract the local features from detected keypoints. Later these features are clustered to produce a visual word dictionary.
In addition, for each shot they produce a histogram of occurrences of visual words using the visual word dictionary. Then, the histograms are grouped, meaning that similar visual entities will be grouped together. Finally, the video summary is produced by extracting the frames that contain the important visual entities. Papadopoulos, Chatzichristofis and Papamarkos (2011) also use the bag of visual words, but the object descriptor used is the SURF (Speeded Up Robust Features) (Bay, Ess, Tuytelaars and Van Gool 2008). Unfortunately, no evaluation has been performed between these descriptors applied to video summarization. Li (2012) analyzes the video content using the SIFT features to produce a static summary. The idea is to summarize the video based on the content complexity and the difference between frames. In order to do this, a video segmentation is applied based on the content complexity. Once the video is segmented, the detected shots are merged based on their similarity. Finally, the keyframes are extracted from the detected merged shots.

The SIFT descriptor has been widely used in computer vision for its ability to handle intensity, rotation, scale variations; this makes it a good descriptor but one disadvantage is its high computational cost.

Temporal information has also been used for video summarization. Lim, Uh and Byun (2011) propose a method that first extracts the keyframes considering the temporal information. Then a Region of Interest (ROI) is estimated for the extracted frames. Finally an image is created by arranging the ROI’s according to their time of appearance and their size. The result is a static summary of the video.

Probabilistic models can also be used. In (Liu, Song, Zhang, Bu, Chen and Tao 2012), a method is proposed to extract key frames by using a maximum a posteriori estimation and therefore produce the video summary. The method uses three probabilistic components: the prior of the key frames, the conditional probability of shot boundaries and the conditional probability of each video frame. Then, the Gibbs sampling algorithm is applied for key frame extraction and finally producing a static video summary.

2.4 Methods Based on Text (Video Annotation)

When this feature is available, it becomes a powerful resource. The text associated to a video should be directly associated to the content of the video (Li, Zhou, Xue, Zha and Yu 2011). Under this presumption, we can infer semantic concepts more easily and
efficiently compared to visual and audio data. Text information can be acquired from various sources: captions, textual overlays, subtitles, etc. However, this resource is not always present. We could have text information for movies, sitcoms, etc., but they are not present in security videos, home videos, etc.

One of the first research articles where text features are used in video summarization is proposed by Chen, Wang and Wang (2009). Their method extracts these text features from speech transcript in order to create concept entities. The method defines 4 principal concept entities: “who”, “what”, “where” and “when”. The method attempts to use these concepts in order to represent the structure of the story as it is assumed that any video can be built up according to the characters, the things, the places, and the time. The method uses a relational graph to detect meaningful shots and relations, which serve as the indices for users. Their approach is particularly good in terms of user satisfaction, since users can not only browse the video efficiently (video summary) but also focus on what they are interested in, via a browsing interface (entity interest).

When no text is present in the video, some methods have tried to create them. Speech recognition is usually used in order to create textual data. For example, in sports videos we can use speech recognition to detect keywords such as: “goal!” , “touchdown!” , “foul!” , etc. This technique has been used in (Babaguchi, Kawai and Kitahashi 2001, Han, Hua, Xu and Gong 2002, Dagtas and Abdel-Mottaleb n.d.).

2.5 Methods Based on Audio

Audio can be used, as explained in Section 2.4, for textual data generation. Audio is usually used for detecting generic sound effects which can be associated to some emotion in the videos. For example, cheers, applauses, laughters, shoutings, etc. And also some particular event: shootings, bomb explosions, etc., as used in (Xiong, Radhakrishnan, Divakaran and Huang 2004), (Tjondronegoro, Chen and Pham 2004). Unfortunately, audio information is not always present, but when it is present, it can give us a resource for finding interesting segments in the video. We can identify the speech segment where the actors gave more emphasis to their speech (shouts, discussions, etc.), or the part of a sport game when something interesting happened (foul, goal, etc.) (Radhakrishnan, Divakaran and Xiong 2004). These methods do not only rely on audio information, they usually use visual information in order to make more analysis and propose more elaborated heuristics for video summarization.
2.6 Final Considerations

In this chapter, we have presented a general overview of some of the principal methods for video summarization. Different methods have been studied, they can use color, audio, text, speech, etc. These methods rely their analysis and the resources they use, according to the type of video they are trying to summarize. The methods have heavily relied only on visual information to summarize generic videos. Another approaches consider the semantic meaning in the videos. Although semantics comprehension is still in continuous investigation, many methods have overcome this problem by delimiting the content of the video to be summarized. In this dissertation, we will base our method on visual information in order to perform a semantic summarization by doing object detection.
Capítulo 3

Theoretical Fundamentals

In this chapter, we present some of the theory and definitions that are needed to understand this dissertation. We review some basic concepts about digital videos and the resources for video summarization, such as visual descriptors and machine learning methods (clustering, classification).

3.1 Digital Video

A digital video is a sequence of images (frames). These frames are displayed in rapid succession at a constant rate, this operation gives the illusion of movement. The more frames we display per second, the smoother the illusion of movement becomes. However, as the video contains more frames its file size increases. Furthermore, a video can not only contain visual information, there can also exist audio and text information that can be synchronized to certain sequence of frames. These facts make a video a complex container of data.

For the sake of our proposal, we will consider a video as a sequence of shots. A shot is defined as an image sequence that presents continuous action which is captured from a single operation of a single camera and its visual content can be represented by keyframes. Shots can be grouped together into a scene. Scenes are composed by a number of shots that share some similarity and their time of appearance is usually close between each other. The anatomy of a video can be seen in Figure 3.1, it also represents the hierarchical internal structure of any video.
3.1.1 Terminology

We now give a formal definition of the terms we use in this dissertation.

- **Video**: A video \( V \) is a sequence of frames \( f_t \) and can be defined as:

\[
V = (f_t)_{t \in [0, t-1]}
\]

where \( t \) is the number of frames.

- **Frame**: A frame (image), has a number of discrete pixels locations and is define as:

\[
f_t(x, y) = (r, g, b)
\]

Where \( x \in 1...M, y \in 1...N, (x, y) \) represent the location of a pixel within an image, \( M \times N \) represent the size of the frame and \( (r, g, b) \) represent the brightness values in the red, green and blue bands respectively.

- **Shot**: A shot \( S \), captures a continuous action from a single camera where camera motion and object motion are permitted. A shot represents a spatio-temporal frame sequence.
• Scene: A scene \( C \), is composed of a certain number of shots that are interrelated and unified by similar features and by temporal proximity. While a shot represents a physical video unit, a scene represents a semantic video unit.

• Feature: In image processing the concept of feature is used to denote a piece of information which is relevant for solving the computational task related to a certain application. We can refer a feature as:

  1. The result of a general neighborhood operation, a feature detector is applied to the image.
  2. An specific structure in the image itself, which can go from simple structures such as points or edges to more complex structures such as objects.

• Keyframe: It is the frame that represents the salient visual content of a shot. Depending on the complexity of the content of the shot, one or more key frames can be extracted.

3.2 Visual Effects in Videos

The production of videos usually involves two important operations, which are: shooting and edition operations (Camara Chavez, Precioso, Cord, Phillip Foligue and de A. Araujo 2007). The shooting operation consist in the generation of the different shots that compose the video. The second operation involves the creation of a structured final video. To achieve this, different visual effects have been added to provide smooth transitions between the shots.

These visual effects usually consist of abrupt transitions (cuts) and gradual transitions (dissolves or fades).

3.2.1 Cut

This is a very straightforward transition that does not involve any complex effect. A cut can be defined as a sharp transition, it is characterized by the abrupt change between consecutive shots. this abrupt change is usually easy to identify.
3.2.2 Fades and Dissolves

Fades and dissolves transitions consist of spreading the boundaries of two adjacent shots by creating new intermediary frames. This means that this type of transition have a starting and an ending frame which identifies the whole transition (Camara Chavez, Precioso, Cord, Phillip Foliguet and de A. Araujo 2007).

**Fade-In**

A fade-in process is characterized by a progressive appearing of a shot. The first frame of the fade-in is a monochromatic frame and it progressively shows the first images of the shot. An example of a fade in effect is shown in Figure 3.2

![Figure 3.2: A fade-in transition](image)

**Fade-Out**

A fade-out effect is characterized by a progressive darkening of a shot until the last frame becomes completely black. An example of a fade-out effect is shown in Figure 3.3

**Dissolve**

The dissolve is characterized by a progressive change of a shot $S_i$ into a shot $S_{i+1}$ with non-null duration. This means that final frames of Shot $S_i$ are combined with the first
Figura 3.3: A fade-out transition

frames of Shot $S_{i+1}$ to create the transition. An example of a fade-out effect is shown in Figure 3.4

Figura 3.4: A dissolve transition

### 3.3 Image Descriptors

A video is a rich container of visual information. This type of information is used by many techniques to segment the video and even to detect and recognize objects and events. In order to summarize a video, the approach relies on the objects contained in the video and their features.

Some of the most popular features are: color, shape, movement, texture, etc. They are extracted and stored in a feature vector for later use.

#### 3.3.1 Color Histogram

A color histogram is a very popular resource in image processing. Images contain color information in a certain color space, such as RGB (Red, Green, Blue), HSV (Hue,
Saturation and brightness Value), etc. A color histogram represents the distribution of colors in an image. A histogram is obtained by splitting the range of the data into equal-sized bins (class-intervals), each bin representing a certain intensity value range.

Let \( f(x, y) \) be a color image (frame) of size \( M \times N \), which consists of three channels \( f = (I_1; I_2; I_3) \). In order to compute a histogram \( h_c(f, b) \), each pixel in the image (frame) \( f \) is analyzed and is assigned to its corresponding \( b \)-th bin depending on the pixel intensity. The final value of a bin is the number of pixels assigned to it. Equation 3.1 gives a formal definition of this procedure.

\[
h_c(f, b) = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \begin{cases} 
1 & \text{if } f(x, y) \text{ in bin } b \\
0 & \text{otherwise}
\end{cases} 
\]

RGB Histogram

In the RGB color space, the RGB histogram is computed by using a combination of three 1-D based histograms, one for each of the RGB color channels (Red, Green and Blue).

Hue Histogram

In the HSV color space, using only the Hue histogram has been very popular. The reason is that the Hue histogram is attributed to give more robust information concerning light intensity, reflecting scale-invariant and shift-invariant (de Sande, Gevers and Snoek 2010, de Sande, Gevers and Snoek 2008).

Using color histograms as image descriptors have several advantages. They are invariant to image rotation and change slowly under the variations of viewing angle and scale (Swain 1993). Another important advantage is its low computational cost, it can be calculated with simple and fast operations.

One important disadvantage is that no spatial or structural information is taken into account. As a consequence, two completely different images can share similar histograms. Pass and Zabih (1999) proposes a joint histogram to deal with this problem, incorporating additional information by selecting a set of local pixel features and constructing a multidimensional histogram. The technique is said to outperform color histograms but we have to consider that this will increase the computational cost.
3.3.2 Projection Histograms

Another resource that has been used as a visual descriptor is projection histograms. A projection can be defined as the operation of mapping an image into a one-dimensional vector. For 2D images, the resulting vector is called the projection histogram. The values of the histogram are the sum of the pixels along a particular direction (Trier, Jain and Taxt 1996). Two types of projection histograms can be defined, horizontal and vertical. Considering an image $f(x, y)$, the horizontal histogram will sum the pixels projected onto the vertical axis $x$ as we can see in Equation 3.2. On the other hand, the vertical projection histogram is the sum of the pixels projected onto the horizontal axis $y$ as shown in Equation 3.3.

$$M_{hor}(y) = \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} f(x, y) d(x) \quad (3.2)$$

$$M_{ver}(x) = \frac{1}{y_2 - y_1} \int_{y_1}^{y_2} f(x, y) d(y) \quad (3.3)$$

3.3.3 Scale-Invariant Feature Transform (SIFT)

SIFT is a popular algorithm in computer vision. It was proposed by Lowe (1999). It is an algorithm useful to detect and describe local features in images, it has been used in many application such as object recognition, robotic mapping, 3D modeling, gesture recognition, video tracking, individual identification.

In video summarization, some methods have based their analysis on object detection and recognition. The objects in the video can appear in any location and can also appear at different scales due to zooming effects. The SIFT descriptor is proposed due to its invariance to:

1. Scale
2. Rotation
3. Illumination
4. Viewpoint
We now present the different parts of the algorithm.

**Construction of the scale space**

In this initial step, internal representations of the original image are created to ensure scale invariance. This is done by generating a “scale space”. To create a scale space, progressively blurred out images are generated from the original image. Then, the original image is resized to half its size, after that more blurred out images are generated again. This process repeats as many times as it is necessary. Each level of resized images are called octaves.

It must be noted that the number of octaves and scales produced have a direct relation to the original image size. The author of SIFT suggests that 4 octaves and 5 blur levels are ideal for the algorithm.

In order to blur the image, a convolution of the gaussian operator and the image is performed. A Gaussian blur operator is applied to each pixel, the final result is a blurred image. Equation 3.4 defines this operation.

\[
L(x, y, \sigma) = G(x, y, \sigma) \otimes f(x, y)
\]  

(3.4)

Where:

- \( L \) is the resulting blurred image.
- \( G \) is the Gaussian Blur operator defined in Equation 3.5.
- \( f \) is the image to be processed.
- \( x, y \) are the location coordinates of the pixel.
- \( \sigma \) defines the amount of blur to be applied.
- \( \otimes \) is the convolution operation.

\[
G(x, y, \sigma) = \frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2}
\]  

(3.5)
Laplacian of Gaussian

In this step, the edges and corners on the image are located, these will be used for finding keypoints. The algorithm could use the Laplacian of Gaussian directly to do this, but it is computationally expensive. To overcome this problem the method proposes a different procedure. They propose to use the scale space previously computed, then calculate the difference between two consecutive scales. This is actually a difference of Gaussians (DoG) as shown in Figure 3.5. The resulting image from the DoG is approximately equivalent to the Laplacian of Gaussian. So the operation becomes a simple difference of images instead of a more complicated procedure.

\[
D(x, y, \sigma) = L(x, y, k_i \sigma) - L(x, y, k_j \sigma) \tag{3.6}
\]

Where \( L(x, y, k\sigma) \) is the result of the convolution operation defined in 3.5.
Locate Potential keypoints

In this step, the possible keypoints using the resulting image explained in Subsection 3.3.3 are located. This process has two parts:

1. Locate maxima/minima in DoG images: this is done by iterating each pixel and checking its corresponding neighbors. The current image and the images above and below it are used. In Figure 3.6, the X marks the current pixel to be processed and the circles mark the neighbors. The evaluated pixel has total of 26 neighbors, the current pixel is marked as a keypoint if it is the greatest of the least of all the 26 neighbors. After doing this, several maxima and minima points are found, but they are an approximation. The real maxima and minima points are located somewhere inside the pixel. To find the actual position the subpixel location needs to be calculated.

![Figure 3.6: Locating maxima/minima in DoG images](image)

2. Subpixel maxima/minima: using the detected pixel data, the subpixel location is computed. This is done by the Taylor expansion of the image around the approximate keypoint. Equation 3.7 defines the Taylor series expansion. This will accurately determine the position of the keypoint

\[
D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x
\]  

(3.7)
Filter Edge and Low Contrast Responses

At this point, some of the several detected keypoints might not be useful. Two things can happen, either some of them are located on an edge, or they may not have enough contrast. If this things happen, then the keypoint is considered to be irrelevant and should be ignored.

To remove a low contrast feature, a threshold $th$ is used. If the magnitude of the intensity in the DoG image is less than $th$, then the keypoint is ignored.

To remove edges, two gradients are calculated at the keypoint. The objective is to search for corners and they are detected when both gradients are big. In order to do this, the Hessian matrix defined in Equation 3.8 is used. If the evaluated keypoint is not detected as a corner, then it is ignored.

$$
M = \begin{pmatrix}
D_{xx} & D_{xy} \\
D_{xy} & D_{yy}
\end{pmatrix}
$$  (3.8)

Assign Keypoints Orientations

In this step, the gradient directions and magnitudes around each detected keypoint are collected. Then, the most prominent orientation in that region is computed. Finally, this orientation is assigned to the keypoint. The size of the region will depend on the image scale.

Generate SIFT features

The final step is to uniquely identify features. In order to do this, an initial 16 window of pixels around the keypoint is taken, then this windows is splitted into 16 $4 \times 4$ windows. An histogram of 8 bins for each of these $4 \times 4$ windows is generated. These bins will correspond to the gradient orientations. For example, the first bin will count the data in the interval $[0, 44]$ degrees, the next bin will count $[45, 89]$ degrees, and so on until the 360 degrees are reached. At this point, there are $4 \times 4$ windows and for each window a 8 bin histogram is generated, the final result will be a vector of size 128. The keypoint is uniquely identified by this feature vector.
3.3.4 A SIFT Descriptor with Color Invariant Characteristics (CSIFT)

One big challenge for a feature descriptor is to be robust to imaging conditions. The descriptor should provide features that are invariant to geometrical variations like rotation, translation, scaling, etc. Many approaches, such as SIFT, use local features to be geometrical invariant, but they avoid using colored images since it can add certain degree of difficulty.

The original SIFT only uses information of gray scale images. However, color can provide important information in object description. Abdel-Hakim and Farag (2006) propose CSIFT, their approach uses a color invariant space to build the SIFT descriptor instead of the traditional gray scale.

CSIFT is a local invariant feature descriptor that allows the descriptor to be more robust to color variations. The CSIFT algorithm uses the same steps as in the SIFT algorithm. The interest points detection is performed by a difference-of-Gaussian for the input image in different scales. The difference is that the working space is the result of a color invariance model.

Geusebroek, Van den Boomgaard, Smeulders and Geerts (2001) propose the color invariance method, which is suitable for modelling non-transparent/translucent materials. To calculate its values from the RGB color space, a Gaussian color model is used. This model represents the spectral information and the local image structure. Abdel-Hakim and Farag (2006) propose to compute the values from the product of two linear transformations. The implementation of the Gaussian color model using the RGB values can be calculated using Equation 3.9.

\[
\begin{pmatrix}
\hat{E} \\
\hat{E}_\lambda \\
\hat{E}_{\lambda\lambda}
\end{pmatrix} =
\begin{pmatrix}
.06 & .63 & .27 \\
.3 & .04 & -.35 \\
.34 & -.6 & .17
\end{pmatrix}
\begin{pmatrix}
R \\
G \\
B
\end{pmatrix}
\]  
(3.9)

where \( \hat{E}, \hat{E}_\lambda, \hat{E}_{\lambda\lambda} \) are the computed color invariants.

Once the interest points are detected, the descriptor building follows the same steps as the SIFT algorithm.
3.3.5 HUESIFT

Weijer, Gevers and Bagdanov (2006) introduced the idea of concatenating the Hue histogram to the SIFT descriptor. Consequently producing the HUESIFT descriptor, which was later used by (de Sande, Gevers and Snoek 2010). Similar to the hue histogram, it shares the same properties. This means that the HUESIFT descriptor is scale-invariant and shift-invariant. However, one important drawback is that, unlike the SIFT descriptor; the HUESIFT is not invariant to illumination color changes. This behavior was noted by de Sande, Gevers and Snoek (2008).

3.3.6 SURF: Speeded-Up Robust Features

Bay, Ess, Tuytelaars and Van Gool (2008) propose SURF (Speeded-Up Robust Features), it is a fast and robust algorithm for local similarity invariant image representation. SURF can be used in object recognition and tracking. Its fast computation makes it suitable for real-time applications.

1. Scale-space approximation via box filters. In order to make the features invariant to scale, a scale-space representation is usually computed using Gaussian convolutions of the image at certain scales. SURF approximates the Gaussian Kernel and spatial derivatives using rectangular functions. The approach treats these functions as box filters. Doing this, makes the convolution complexity linear and therefore faster. The resulting scale-space representation using box filters is very similar to the one generated by Gaussian kernels. As in Gaussian kernels, a scale parameter $\sigma$ is defined to create different scales of blurred images.

2. Interest points detection:

To detect the interest points, SURF follows three steps:

- Feature detection. To detect scale-invariant features a scale-normalized second order derivative on the scale space representation is used. SURF approximates this using a scale-normalized determinant of the Hessian (DoH) operator.

- Feature selection. Once the DoH operator has been applied to the scale-space representation of the image, the local maxima is selected. To select them, a neighborhood of $3 \times 3 \times 3$ values are compared in every point of the scale space. In order to detect the most salient features from the set of detected local
maxima points, the algorithm uses a threshold $t_H$ on the response of the DoH operator. This is applied to every local maxima point, and a set of interest point candidates is generated.

- Scale-space location refinement. In order to refine the location of the interest points, a second order interpolation is used.

- Storage of the interest points. The interest points detected are saved with information about their position $(x, y)$ and their respective $\sigma$.

3. Local descriptors construction. Equation 3.10 formally describes the interest points detected.

$$\{M_i : (x_i, y_i, \sigma_i)\}_{i=1,2,...,N}$$ (3.10)

For each interest point $M_i$, the local orientation is computed from the local distribution of the gradient orientation, which is obtained by the convolution with box filters. An orientation $\theta_i$ is computed.

To construct the SURF descriptor, for each interest point a $16 \times 4$ vector is computed. The vector represents normalized gradient statistics corresponding to the scaled and oriented neighborhood of the interest point.

### 3.3.7 Space-Time Interest Points (STIP)

Laptev (2005) proposes an algorithm to detect spatio-temporal events using space-time interest points. The features contain not only space but also time information. Interest points are detected using the Harris operator. The algorithm looks for structures with significant local variations in terms of space and time.

#### Spatio-Temporal Interest Point Detection

To detect interest points in the spatial domain, an image can me modeled by a linear scale-space representation. The image can be defined by $f^{sp} : R^2 \rightarrow R$. A function $L^{sp}$ is used to represent the linear scale-space, where $L^{sp} : R^2 \times R_+ \rightarrow R$ is defined in Equation 3.11
The theoretical fundamentals of interest point detection and spatio-temporal image analysis are explored in this section. The Harris operator is applied to detect interest points in a spatial domain. The idea is to find locations where the feature map $f_{sp}$ shows significant changes in both spatial directions. To do this, a second moment descriptor $\mu_{sp}$ is used. Then, the algorithm calculates the eigenvalues $\lambda_1, \lambda_2$ (where $\lambda_1 \leq \lambda_2$) of $\mu_{sp}$. If $\lambda_1, \lambda_2$ have large values, then an interest point is detected. Equation 3.13 defines the operator to obtain the Harris interest points.

$$H^{sp} = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

(3.13)

where $k$ is a constant, in the literature the value used is usually $k = 0.04$.

To detect interest points in the spatio-temporal domain, Laptev (2005) proposes an operator that is sensitive to events in temporal image sequences at specific locations considering space-time information. The interest points detected will represent values in local spatio-temporal volumes, they are required to have certain variation along the spatial and the temporal directions.

A function $f : R^2 \times R \rightarrow R$ is used to model the spatio-temporal image sequence. The algorithm generates the linear scale-space representation $L : R^2 \times R \times R^2 \rightarrow R$, using a Gaussian kernel to convolve $f$. The Gaussian kernel has spatial variance $\sigma_t^2$ and temporal variance $\tau_t^2$. This representation is defined in Equation 3.14.

$$L(x, y, t; \sigma_t^2, \tau_t^2) = g(x, y, t; \sigma_t^2, \tau_t^2) * f(x, y, t)$$

(3.14)

where the Gaussian kernel $g$ is defined in Equation 3.15.
As in the spatial domain, a spatio-temporal second moment matrix is used. The matrix is composed by a first order spatial and temporal derivatives, these values are averaged using a Gaussian function \( g(x, y, t; \sigma_i^2, \tau_i^2) \).

\[
\mu = g(x, y, t; \sigma_i^2, \tau_i^2) \ast \begin{pmatrix}
L_x^2 & L_xL_y & L_xL_t \\
L_xL_y & L_y^2 & L_yL_t \\
L_xL_t & L_yL_t & L_t^2
\end{pmatrix}
\]  

(3.16)

The algorithm uses the integration scales \( \sigma_i^2, \tau_i^2 \), which are related to the local scales \( \sigma_i^2, \tau_i^2 \), where \( \sigma_i^2 = s\sigma_i^2, \tau_i^2 = s\tau_i^2 \), and \( s \) represents the scale.

Then, the eigenvalues \( \lambda_1, \lambda_2, \lambda_3 \) of \( \mu \) are calculated. (Laptev 2005) proposes to extend the Harris corner function. The extended function is defined in Equation 3.17.

\[
H = \lambda_1\lambda_2\lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3
\]  

(3.17)

where the constant \( k \) is usually set to the value \( k \approx 0.005 \).

Finally, the local positive spatio-temporal maxima in \( H \) are detected, the results will be the spatio-temporal interest points of \( f \).
3.3.8 Histograms of Oriented Gradients (HoG)

Dalal and Triggs (2005) propose the Histogram of Oriented Gradient descriptors. The basic idea of this descriptor is that using the distribution of intensity gradients or edge directions, the local object appearance and shape can be described.

The algorithm first creates cells by dividing the image into small connected regions, each cell will have a set of associated pixels within it. Then, for each of the cells the algorithm computes a histogram of gradient directions or edge orientations. The descriptor is formed by the combination of these histograms. To further improve the descriptor, a normalization can be performed to provide the descriptor a better invariance to changes in illumination. To do this normalization, the algorithm takes a block which is a larger region of the image and measures the intensity. The obtained value is used to normalize all the local histograms that belong to cells that are within the block.

The algorithm is further described below:

**Gamma/Colour Normalization**

This is a pre-processing step. A global image normalization is performed in order to reduce the influence of illumination effects.

**Gradient Computation**

In this step, the gradient values are calculated. Dalal and Triggs (2005) propose to apply a simple 1-D centered mask, which according to their experiments gave the best results. The used masks are: \([-1, 0, 1]\) and \([-1, 0, 1]^T\). More complex masks, such as \(3 \times 3\) sobel masks where evaluated by the authors but the results obtained with them are not as good as the ones obtained with the simple 1-D masks.

**Orientation binning**

At this stage, the algorithm generates cell histograms. The image is divided in small spatial regions, these regions are called “cells”. For each cell, a local 1-D histogram of gradients is calculated for each cell. Based on the values obtained in the gradient computation, each pixel in the cell casts a weighted vote for an orientation-based histogram. The
weight can be computed from the gradient magnitude, or some function of the magnitude (square, square root, clipped). Nevertheless, gradient magnitude generally produces the best results according to the authors. Cells can either be of rectangular or radial shape.

**Normalization and Descriptor Blocks**

Normalization gives the descriptor better invariance to illumination, shadowing, and edge contrast. To do this, local groups of cells called “blocks” are normalized. The algorithm accumulates a measure of local histogram “energy” over the blocks, the value calculated from this is used to normalize the different cells that belong to the block.

To normalize a block, let $v$ be the non normalized vector containing the histograms in a block, $\|v\|_k$ be its $k$—norm for $k = 1, 2$ and $\epsilon$ a constant. The normalization can be calculated using several schemes such as: $L2 – norm$, $L1 – norm$, $L1 – sqrt$. Equations 3.18, 3.19 and 3.20 defines each of these normalizations respectively.

- **$L2 – norm$**

\[
v \rightarrow \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}} \tag{3.18}
\]

- **$L1 – norm$**

\[
v \rightarrow \frac{v}{\|v\|_1 + \epsilon} \tag{3.19}
\]

- **$L1 – sqrt$**

\[
v \rightarrow \sqrt{\|v\|_1 + \epsilon} \tag{3.20}
\]

The authors found that normalizations $L2 – norm$ and $L1 – sqrt$ provide similar performance, and $L1 – norm$ provided less reliable performance.

Another issue to have in mind is that, a cell can be shared between blocks. But the normalization procedure is block dependant and therefore unique. Consequently, a cell may be present several times in the final output vector with different normalized values. The normalized block is denominated the histogram of oriented gradients (HoG) descriptor.
Finally, the algorithm collects the HoG descriptors from all the blocks covering the detection window into a integrated feature vector.

3.4 Classification

Classification is one of the most frequent tasks to be solved when taking decisions. The problem appears when certain object needs to be associated to some class. To do this, the models base their analysis in the features and possible relations of the object. There exist two types of classification: supervised classification and unsupervised classification.

In the supervised classification, some previously labeled patterns are at hand. For these patterns, the classes they belong to are already known. The problem is to classify new unlabeled patterns. Therefore, the labeled patterns help us to train the classification model, once the model has been trained it can be used to classify new objects. The problem is that in real life, that previous information is not always present.

The unsupervised classification is useful when no previous class information is present. This type of classification is also known as clustering. The clustering model is widely used in video summarization. The patterns are feature vectors that represent multidimensional points, the clustering models use these feature vectors to compute the similarity between them and thus create and classify the patterns in classes.

3.4.1 K-means

The K-means has been one of the most popular unsupervised clustering algorithms, it was proposed by MacQueen (1967). The procedure needs to know \( a \ priori \) the number of clusters \( k \), a cluster consists of feature vectors with similar characteristics that is represented by a centroid. Initially, \( k \) centroids are located at random positions.

Then, each feature vector of the data set is associated to the nearest centroid. The next step, consists on re-calculating the centroids positions. The data set has to be tied to the new centroids locations. These operations will be repeated until reaching the point where the centroids no longer change positions. This algorithm aims to minimize an objective function defined in Equation 3.21, in this case a squared error function.
\[ j = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2 \]  \tag{3.21}

where \( \| x_i^{(j)} - c_j \|^2 \) represents the distance between the element \( x_i^{(j)} \) and the centroid \( c_j \).

Unfortunately, the K-means algorithm will not always find the most optimal classification. Another disadvantage is its sensitiveness to the initial centroids position. Consequently, the algorithm will not always produce the same result for a data set.

### 3.4.2 X-means: Extending K-means with Efficient Estimation of the Number of Clusters

The original K-means algorithm has been very popular due to its simplicity and quick implementation. However, the K-means algorithm has three shortcomings that are overcome by X-means algorithm (Pelleg and Moore 2000). First, it can be slow due to repetitive interactions. Second, the number of clusters has to be known \textit{a priori}, meaning the user has to input the quantity of classes. Finally, Pelleg and Moore (2000) stated that when a fixed value \( K \) is used, the algorithm finds worse local optima compared to results obtained when the value \( K \) is dynamically modified.

Pelleg and Moore (2000) propose the X-means algorithm to quickly estimate the \( K \) number of clusters. The user is required to input a reasonable range of values of \( K \), minimum and maximum. Then, the algorithm uses this range to select the best set of centroids and the best value for \( K \). The model selection criterion used is the Bayesian Information Criterion (BIC).

The definitions used are:

- \( \mu_j \) coordinate of the \( j \)th centroid.
- \( i \) the index of the centroid that is closest to the \( i \)th data point.
- \( D \) the data set.
- \( R = |D| \).
- \( M \) number of dimensions.
- \( \Sigma = \text{diag}(\sigma^2) \) Gaussian Covariance Matrix.
The X-means algorithm consists of the following operations:

**X-means:**

1. **Improve Parameters**
2. **Improve Structure**
3. If $K > K_{\text{max}}$ Stop and generate the best scoring model.
   Else Goto Step 1

These operations are further explained below:

1. **Improve Parameters.** This is a simple operation, the algorithm executes a conventional $K$–means.

2. **Improve Structure.** This operation is the core of the algorithm. It finds out if new centroids should be created and also defines the location of the new centroids.

   The strategy is the following:
   - Split each centroid (parent) into two children
   - They are moved to a distance that is proportional to the size of the region in the opposite direction.
   - Each parent has a region, a local $K$–means is executed with $K = 2$.
   - A model selection (BIC) is executed on the recently created pairs of children. The model helps the algorithm decide whether to create new centroids or not. Given the data $D$ and a set of alternative models $M_j$ (these models are the solutions with different values of $K$). The BIC score can be calculated using 3.22.

   $BIC(M_j) = \hat{l}_j(D) - \frac{p_j}{2} \cdot \log R$  \hfill (3.22)

   where $\hat{l}_j(D)$ is the log-likelihood of the data according to the $j$–th model. $p_j$ is the number of parameters in $M_j$, it is the sum of $K - 1$ class probabilities. The $R$ value is replaced with the total number of points which belong to the centroids under evaluation.
• All stable-state information is saved. Therefore, the algorithm no longer recomputes stable values, reducing the number of calculations in each iteration.

The main contribution is to incorporate the model selection into their algorithm. Using statistically-based criteria to make local decisions helps the algorithm find the optimal $K$ value.

### 3.5 The Bag-of-Words model

The bag-of-words model (BoW), was originally proposed for text processing. It was applied to simplify the representation used in natural language processing and information retrieval. In computer vision, the model was adapted by considering image features as words. To represent an image using BoW model, an image can be treated as a document. Three operations are executed: Feature detection, feature description and codebook generation.

• Feature representation: Feature detection and feature description. This is usually done by applying a keypoint detection algorithm and a keypoint descriptor. One of the most popular descriptors is the SIFT descriptors. The result from this operation is a set of feature vectors, one for each detected keypoint.

• Codebook generation It is usually achieved by performing K-means clustering over all the vectors. Codewords are then defined as the centers of the learned clusters. The number of the clusters is the codebook size.

After performing these operations, the visual word vocabulary is defined. Then, future images can be described by the occurrences of the visual words according to our visual word vocabulary. This is done by computing a histogram of occurrences (visual word vectors). In 3.7, a general overview of the Bag-of-Words approach is presented.

### 3.6 Final considerations

In this chapter, we have reviewed some fundamental concepts that are used in this dissertation. The covered concepts were about visual descriptors and classification. We have presented some visual descriptors that can be used for video summarization
considering semantic and non-semantic information. Different models rely heavily on clustering algorithms to perform summarization by similarity.

Figura 3.7: The Bag-of-Words model
Capítulo 4

Proposed Method

In this chapter, a method for static video summarization is described. The presented method is used in this dissertation to generate the different video summaries. Additionally, an evaluation model is reviewed to estimate the quality of the generated summaries. The chapter is organized as follows: in Section 4.1 each of the steps for summarizing videos is described. In Section 5.3, the model for evaluating video summaries is presented.

4.1 Proposed Method for Summarizing Videos

In Figure 4.1, we present a general overview of our proposed method. Initially, a temporal segmentation procedure is executed. As a result, several shots are detected. Then, each shot is clustered. This is done to detect frame samples from each shot. Later, for each of the frames previously detected, a feature description procedure is applied. Afterward, a Bag-of-Words (BoW) approach is adopted. The detected local features are clustered to generate our Visual Word Vocabulary. Next, the histograms of occurrence of visual words are created for each frame of the detected frame samples. The histograms of occurrence are clustered, the method finds the frames that are near to each cluster’s centroid. The frames that represent the centroids are considered as keyframes. The method filters the results to eliminate possible redundant keyframes. Finally, the keyframes are ordered in chronological order. The final result is the video summary.
4.1.1 Temporal Video Segmentation

Video segmentation is usually the first procedure carried on by many methods in the literature. It is very important to segment the video to analyze only the segments that can contain important information. In our proposed method, it is important to deal only with frames with significant and valid information. To accomplish this, our method first computes the color histograms for each pair of adjacent frames in the video. Then, the cosine similarity measure is computed between two consecutive histograms, the resulting information is a similarity vector. The method analyzes this information to detect the boundaries of the segment, this is done by examining and detecting abrupt changes in the similarity vector. When two adjacent frames have a high dissimilarity, an abrupt change of images is detected. This implies that a new shot has been detected. Since the dissimilarity information is more useful, the similarity vector data is inverted. High values of dissimilarity will give us information about the shot boundaries in the video. Empirically, we found that using a threshold $th$ of value 0.4 is good to detect abrupt changes.
In Figure 4.2, a dissimilarity vector is shown for a video. As we can see, using \( th = 0.4 \) will detect several false shot cuts. In order to overcome this problem, a procedure is executed to refine the dissimilarity vector.

![Graph showing dissimilarity vector](image)

**Figura 4.2:** A dissimilarity vector computed from video *HCIL Symposium 2002 - Introduction, segment 01*

The procedure works as follows: each of the values in the dissimilarity vector is analyzed. Each value becomes a pivot when evaluating the dissimilarity vector. Then, a neighborhood of values around the pivot is taken, excluding the value of the pivot. The maximum value \( mv \) of the neighborhood is calculated. If the value of the pivot \( v \) is greater than \( mv \), the value of the pivot is modified by applying Equation 4.1.

\[
s_i = \begin{cases} 
\frac{v_i - mv_i}{v_i} & \text{if } v_i > mv_i \\
\frac{v_i}{v_i} & \text{otherwise}
\end{cases}
\]  

(4.1)

where \( i \) is a position in the dissimilarity vector, \( s_i \) is the computed value, \( v_i \) is the value of the pivot and \( mv_i \) is the maximum value of the neighborhood around the pivot.

After applying Equation 4.1 the values in the dissimilarity vector are refined. Figure 4.3 shows the refined dissimilarity vector for a video, the marked rounds in the peaks are the recognized abrupt changes.
To know the number of segments present on the video, the method simply counts the number of abrupt changes.

This type of segmentation is very effective and not computationally expensive. However, there is one shortcoming. Videos usually have visual effects such as fade in, fade out and dissolves. Figure 4.4 shows a dissolve effect.

A dissolve effect can produce distorted images, these type of images have a great impact in the discriminative power of a descriptor. An frame that is produced during the dissolve transition will produce a lot of spurious keypoints. These false keypoints will eventually affect the summarization process. To overcome this problem, the method identifies the portions of the video where this effect happens. Once these portions are detected, they are excluded of any posterior analysis.

To detect the possible dissolve effect in the video, our method first compute the variance in each frame of the video and put it in a vector. Figure 4.5 shows the resulting variance vector for a video. Dissolve effects are located in the valley areas in the variance vector. Therefore, the method uses a procedure based on (Won, Chung, Kim, Choi and Park 2003, Camara Chavez, Precioso, Cord, Phillip Foliguet and de A. Araujo 2007) to detect the valleys areas in the variance vector and consequently find the portion of video where a dissolve effect occurs.

At this point, the method has detected the shot boundaries and therefore segmented the video. Furthermore, by detecting the dissolve effects the method can exclude not relevant portions of video that can actually affect the performance of the summarization method.

4.1.2 Detection of Representative Frames

Once the video has been segmented, the method uses the valid parts of the video for analysis. The next step consists of extracting the features. However, the feature extraction process for all the frames can be computationally expensive and moreover, detecting features in contiguous frames can produce repetitive information. Our method applies the X-means algorithm for each portion of the previously segmented video using the color histograms information. This is done to detect the most representative frames for each portion of the analyzed video. All the posterior analysis will be performed using these frames.
4.1.3 Bag of Words (BoW)

Our method uses the Bag-of-Words (BoW) approach to summarize videos. To adopt the approach, an image can be considered as a document.

The “words” are the visual entities found in the image. They will describe the object and therefore represent our semantic entities. Using their information we can perform a semantic summarization based on the objects of the video.

The Bag-of-Words (BoW) approach consists of three operations: feature detection, feature description and visual word vocabulary generation. Many local descriptors, such as SIFT, can be used for feature detection and description. A visual word vocabulary is generated from the feature vectors obtained during the feature detection process, each visual word (codeword) represents a group of several similar features. The visual word vocabulary defines a space of all entities occurring in the video, it can be used to semantically summarize the video based on the entities present on it.

Feature Detection and Description

Images can be described by global features, such as: color, texture, etc. But they can also be described by the objects contained within it, using local descriptors. A feature detection and description algorithm can be computed on an image to detect and describe the interest points of the objects. Such algorithm must be robust to geometrical transformations, noise and illumination variances.

For this dissertation, the local descriptor extracts the features for each of the frames detected in the previous step. The method can use any local descriptor, such as:

- SIFT
- SURF
- CSIFT
- HOG
- HUESIFT

In order to extract semantic information, a visual word vocabulary is created from the features of the selected frames.
Visual Word Vocabulary

A word vocabulary, also known in the literature as the “codebook”, defines the “codewords” used in the Bag-of-Words. A “codeword” represents a group of feature vectors that share similar characteristics.

To compute the “codewords” all feature vectors are grouped and classified using a clustering algorithm. Any clustering algorithm, such as K-means, can be used. Then, the “codeword” is defined as the center of the detected cluster.

The “codebook” size establish the number of visual words. This value determines the number of clusters to be found. For this dissertation, the “codewords” are the visual words and the “codebook” is the entire visual word vocabulary. Correctly establishing the number of visual words is still a open problem.

4.1.4 Histogram of Visual Words

A histogram of visual words is created by counting the occurrence of the visual words. For each representative frame, the local image features are used to find the visual words that occur in the image. These occurrences are counted and arranged in a vector. Consequently, each representative frame will have an associated vector of visual word occurrences (visual word vector).

4.1.5 Visual Word Vectors Clustering

Finally, the method uses all the visual word vectors recently obtained and applies the X-means algorithm. Frames with similar visual entities are grouped together. Then, for each cluster, the nearest frame to the centroid is chosen as the “keyframe”. All the detected “keyframes” are ordered according to their time of appearance and they represent the video summary or storyboard. By doing it, we have grouped together the most representative frames of the video taking into consideration the semantic information (visual entities) contained in them. This ensures that the final video summary contains the most important visual entities present in the video.
4.1.6 Filter Results

This final operation tries to eliminate possible duplicated “keyframes”. A pairwise similarity measure is calculated from color histograms of consecutive keyframes. Then, keyframes with high similarity are removed from the final video summary. The method uses a threshold of value 0.5 to define a high similarity between two color histograms, as in (de Avila, ao Lopes, da Luz and de Albuquerque Araújo 2011).

4.2 Final Considerations

In this chapter, a method for static video summarization taking into consideration the semantic information has been presented. We have also cited the local descriptors we will later use in our experiments. As we can see in the next chapter, the proposed method and the local descriptors we use lead into promising video summaries.
**Proposed Method**

**Figura 4.3:** A refined dissimilarity vector computed from video *HCIL Symposium 2002 - Introduction, segment 01*

**Figura 4.4:** Dissolve effect in video *HCIL Symposium 2002 - Introduction, segment 01* between frames 2139 - 2162
**Figura 4.5:** Variance vector for video *HCIL Symposium 2002 - Introduction, segment 01*. The circles are the detected dissolve effects.
Capítulo 5

Experimental Results

In this chapter, the performed experiments and the results obtained using the proposed method are presented. In Section 5.1, the used data set is described. In Section 5.2, the parameter values used for the experiments are defined. Later, in Section ?? the evaluation methods used to measure the obtained results are described. Finally, in Section 5.4, the different experiments are shown.

5.1 Data Set

The whole data set used for experiments is available for download through the Open Video Project (O.V. Project). The O.V. Project is sponsored by and developed at the School of Information and Library Science, University of North Carolina Chapel Hill. The O.V. Project basic framework began to work in 1988 and the initial data set was also available the same year with 195 video segments. Over the years, additional video segments have been added. Some of the contributors are the CMU Informedia Project, the Howard Hughes Medical Institute, and the Prelinger Archives.

O. V. Project mission is to collect and make available a repository of video contents for the research community (multimedia retrieval, digital library, etc.). Currently, the O.V. Project provides around 4000 videos, available in different formats (MPEG-1, MPEG-2, MPEG-4, and QuickTime) and in different genre characteristics (student television, anthropological footage, technology demonstrations, etc). The variety of the videos allows researchers to study a wide range of problems like automatic segmentation, summarization, video content description, etc. Furthermore, the Open Video data set
can be used as a test set, enabling different methods to be compared based in the same data set.

In this dissertation, 50 videos from the O. V. Project have been used. More precisely, they are the same 50 videos de Avila, ao Lopes, da Luz and de Albuquerque Araújo (2011) have used in their experiments. Using the same data set makes it possible to compare both results. The used videos have the MPEG-1 format, their resolution is of $352 \times 240$ pixels with a mean of 30 frames per second. The duration of the videos varies from 1 minute to 4 minutes approximately. They are colored videos with audio information, although this last resource is not used in this dissertation.

In Table 5.1, the information about all the videos used in our experiments is listed. Information about their code, the original video name, the total number of frames and the total duration in minutes are informed.

In addition, in order to evaluate other types of videos containing different genres such as cartoons, news, sports, commercials, tv-shows, home videos, etc. Another set of 50 videos is also used, these videos were extracted from websites like Youtube, and are also available for downloading. These videos were also used by de Avila, ao Lopes, da Luz and de Albuquerque Araújo (2011) in their evaluation.

<table>
<thead>
<tr>
<th>Code</th>
<th>Video Name</th>
<th>Frames</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>614</td>
<td>v21 The Great Web of Water, segment 1</td>
<td>3.279</td>
<td>01:50</td>
</tr>
<tr>
<td>615</td>
<td>v22 The Great Web of Water, segment 2</td>
<td>2.118</td>
<td>01:11</td>
</tr>
<tr>
<td>670</td>
<td>v23 The Great Web of Water, segment 7</td>
<td>1.745</td>
<td>00:59</td>
</tr>
<tr>
<td>684</td>
<td>v24 A New Horizon, segment 1</td>
<td>1.806</td>
<td>01:01</td>
</tr>
<tr>
<td>685</td>
<td>v25 A New Horizon, segment 2</td>
<td>1.797</td>
<td>01:00</td>
</tr>
<tr>
<td>686</td>
<td>v26 A New Horizon, segment 3</td>
<td>6.249</td>
<td>03:29</td>
</tr>
<tr>
<td>687</td>
<td>v27 A New Horizon, segment 4</td>
<td>3.192</td>
<td>01:47</td>
</tr>
<tr>
<td>689</td>
<td>v28 A New Horizon, segment 5</td>
<td>3.561</td>
<td>01:59</td>
</tr>
<tr>
<td>690</td>
<td>v29 A New Horizon, segment 6</td>
<td>1.944</td>
<td>01:05</td>
</tr>
<tr>
<td>661</td>
<td>v30 A New Horizon, segment 8</td>
<td>1.815</td>
<td>01:01</td>
</tr>
</tbody>
</table>

Continued on next page
<table>
<thead>
<tr>
<th>Code</th>
<th>Video Name</th>
<th>Frames</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>663</td>
<td>v31 A New Horizon, segment 10</td>
<td>2.517</td>
<td>01:24</td>
</tr>
<tr>
<td>635</td>
<td>v32 Take Pride in America, segment 1</td>
<td>2.691</td>
<td>01:30</td>
</tr>
<tr>
<td>637</td>
<td>v33 Take Pride in America, segment 3</td>
<td>3.261</td>
<td>01:49</td>
</tr>
<tr>
<td>4586</td>
<td>v34 Digital Jewelry: Wearable Technology for Every Day Life</td>
<td>4.204</td>
<td>03:00</td>
</tr>
<tr>
<td>4923</td>
<td>v35 HCIL Symposium 2002 - Introduction, segment 1</td>
<td>2.336</td>
<td>01:18</td>
</tr>
<tr>
<td>422</td>
<td>v36 Senses And Sensitivity, Introduc. to Lecture 1 presenter</td>
<td>4.221</td>
<td>02:20</td>
</tr>
<tr>
<td>425</td>
<td>v37 Senses And Sensitivity, Introduc. to Lecture 2</td>
<td>3.411</td>
<td>01:53</td>
</tr>
<tr>
<td>430</td>
<td>v38 Senses And Sensitivity, Introduc. to Lecture 3 presenter</td>
<td>4.566</td>
<td>02:32</td>
</tr>
<tr>
<td>538</td>
<td>v39 Senses And Sensitivity, Introduc. to Lecture 4 presenter</td>
<td>5.249</td>
<td>02:55</td>
</tr>
<tr>
<td>719</td>
<td>v40 Exotic Terrane, segment 1</td>
<td>2.94</td>
<td>01:38</td>
</tr>
<tr>
<td>720</td>
<td>v41 Exotic Terrane, segment 2</td>
<td>2.776</td>
<td>01:32</td>
</tr>
<tr>
<td>721</td>
<td>v42 Exotic Terrane, segment 3</td>
<td>2.676</td>
<td>01:29</td>
</tr>
<tr>
<td>722</td>
<td>v43 Exotic Terrane, segment 4</td>
<td>4.797</td>
<td>02:40</td>
</tr>
<tr>
<td>724</td>
<td>v44 Exotic Terrane, segment 6</td>
<td>2.425</td>
<td>01:21</td>
</tr>
<tr>
<td>726</td>
<td>v45 Exotic Terrane, segment 8</td>
<td>2.428</td>
<td>01:21</td>
</tr>
<tr>
<td>731</td>
<td>v46 America’s New Frontier, segment 1</td>
<td>3.591</td>
<td>01:59</td>
</tr>
<tr>
<td>733</td>
<td>v47 America’s New Frontier, segment 3</td>
<td>2.166</td>
<td>01:12</td>
</tr>
<tr>
<td>734</td>
<td>v48 America’s New Frontier, segment 4</td>
<td>3.705</td>
<td>02:03</td>
</tr>
<tr>
<td>737</td>
<td>v49 America’s New Frontier, segment 7</td>
<td>3.615</td>
<td>02:00</td>
</tr>
<tr>
<td>740</td>
<td>v50 America’ ps New Frontier, segment 10</td>
<td>4.83</td>
<td>02:41</td>
</tr>
<tr>
<td>744</td>
<td>v51 The Future of Energy Gases, segment 3</td>
<td>2.934</td>
<td>01:37</td>
</tr>
<tr>
<td>746</td>
<td>v52 The Future of Energy Gases, segment 5</td>
<td>3.615</td>
<td>02:00</td>
</tr>
<tr>
<td>750</td>
<td>v53 The Future of Energy Gases, segment 9</td>
<td>1.884</td>
<td>01:02</td>
</tr>
<tr>
<td>753</td>
<td>v54 The Future of Energy Gases, segment 12</td>
<td>2.886</td>
<td>01:36</td>
</tr>
<tr>
<td>785</td>
<td>v55 Oceanfloor Legacy, segment 1</td>
<td>1.74</td>
<td>00:58</td>
</tr>
<tr>
<td>786</td>
<td>v56 Oceanfloor Legacy, segment 2</td>
<td>2.325</td>
<td>01:17</td>
</tr>
</tbody>
</table>

Continued on next page
5.2 Experimental setup

The experiments have been executed on a machine with the following technical features:

- Operative System: Windows 7
- Processor: Intel Core 2 Duo P8400 2.26GHz
- Memory: 4 GB.
- Programming Language: Matlab

In order to discover the values of the parameters, we have used 10 videos from the O.V. data set. These values were used for the experiments executed using the O.V. data.
set and the Youtube data set. We have discovered and used the following parameters for our experiments:

The X-means algorithm is used at two stages. First, to extract the representative frames for each part of our segmented video. The X-means has two important parameters: the initial number of clusters $K_{\text{initial}}$ and the maximum number of possible clusters $K_{\text{max}}$. During our experiments we discovered that using $K_{\text{initial}} = 5$ and $K_{\text{max}} = 10$, we obtained the best results. This means that each segment will have at least 5 and at most 10 representative frames. The reason behind this, is that no shot has more than 10 representative frames. Second, to generate the video summary, grouping the similar visual entities. To perform this, we have set $K_{\text{initial}} = n_s$ the number of shots detected during the segmentation step ($n_s$) and $K_{\text{max}} = 3 \times n_s$.

To adopt the Bag-of-Words approach we need to set the size of the “codebook” (number of words). The number of visual words used for the experiments is 400. In order to obtain this value, we have performed experiments with 12 videos from the O.V. data set. The best results were obtained using 400 visual words.

We have used several local descriptors for our experiments, such as: SIFT: this dissertation uses the implementation of Vedaldi and Fulkerson (2008), the non-edge selection threshold value is set to 30. The peak selection threshold is set to 2.5. The number of levels per octave of the DoG scale space is set to 2. SURF: we use the implementation of Strandmark (2010). Color descriptors such as: CSIFT, HUESIFT. We use the implementation of de Sande (2010) Finally, the HOG descriptor is also applied using the implementation of Xiankai (2012). For all these descriptors, the default parameters were used.

5.3 Video Summary Evaluation

The resulting video summary needs to be evaluated in order to verify the relevance of the selected “keyframes”, and therefore evaluate how good the method performs.

Unfortunately, despite years of investigation, there is no standard evaluation framework for video summaries. Over the years, different methods have been proposed for video summarization. Nonetheless, they usually define and use their own evaluation framework. Furthermore, the data set used by the methods is, most of the times, not available for downloading. Consequently, there is little comparisons made between methods.
In this dissertation, the evaluation method proposed by de Avila, ao Lopes, da Luz and de Albuquerque Araújo (2011) is adopted. The benefits gained from adopting their evaluation method are:

- The data set is available for downloading.
- There is a ground-truth data set for evaluating the results.
- The obtained results can be compared to the results of other methods using the same data set.
- To reduce the subjectivity in the evaluation task.
- To quantify the summary quality.

The obtain the ground-truth summaries, all video summaries were built manually by a number of users. The users were asked to choose what they thought were the most relevant frames from the video. The resulting summaries from the users are taken as the ground-truth and can therefore be compared to the summaries obtained through the proposed method.

In order to keep the original opinion of the user, CUS makes a comparison between the user summary and the automatic summary. The idea is to take a keyframe from the user summary and a keyframe from the automatic video summary. Then, they are both converted to the HSV color space. Afterwards, a 16 bins color histogram using only the Hue component is computed for both images. A distance is calculated between the two histograms. If the result is superior than a predetermined threshold \( d \), then both are considered as matched frames. If both frames are matched, then both are removed from future iterations. The threshold value used is \( d = 0.5 \), it was the same threshold used by de Avila, ao Lopes, da Luz and de Albuquerque Araújo (2011). Then, the relevance of the summary is measured by two metrics, \( CUS_A \) and \( CUS_E \).

\( CUS_A \) measures the accuracy of the summary, while \( CUS_E \) measures the error rate. They are both defined in Equations 5.1 and 5.2 respectively.

\[
CUS_A = \frac{n_{mAS}}{n_{US}} \tag{5.1}
\]

where \( n_{mAS} \) is the number of matching keyframes from the automatic summary and \( n_{US} \) is the number of keyframes from user summary.
Experimental Results

\[ CUS_E = \frac{n_{\text{mAS}}}{n_{US}} \]  

(5.2)

where \( n_{\text{mAS}} \) is the number of non-matching keyframes from the automatic summary and \( n_{US} \) is the number of keyframes from user summary.

\( CUS_A \) values vary in a range of 0 to 1. The lowest value zero, means that no keyframes between the automatic summary and the user summary were matched, while the highest value one, means that all the keyframes were matched (best case scenario).

\( CUS_E \) values can also vary from 0 to 1. The zero value means that all the keyframes were matched. The value one, is the worst case scenario meaning that none of the keyframes were matched between the two summaries.

5.4 Experiments

All the tables show the mean accuracy rate for \( CUS_A \) and \( CUS_E \). Furthermore, to measure the similarity between summaries we have used 4 different distances: manhattan, cosine similarity, histogram intersection and \( \chi^2 \).

The following local descriptors have been used: HoG(Dalal and Triggs 2005), SURF(Bay, Ess, Tuytelaars and Van Gool 2008), SIFT(Lowe 1999), CSIFT(Abdel-Hakim and Farag 2006) and HUESIFT(de Sande, Gevers and Snoek 2010).

We have compared our results to other methods that use global descriptors, such as: VSUMM (de Avila, ao Lopes, da Luz and de Albuquerque Araújo 2011), DT (Mundur, Rao and Yesha 2006), STIMO (Furini, Geraci, Montangero, Pellegrini, a1, a2 and a3 2010) and the summaries provided by the Open Video Project (OV).

5.4.1 Results for the Open Video Database

In Table 5.2, we show the performance of the visual words using local descriptors and temporal segmentation for the OV dataset. Table 5.3 shows the results obtained with visual words using local descriptors and without using temporal segmentation, and we also show the results obtained using global descriptors.
Experimental Results

As we can see in Table 5.2, local descriptors had a better performance compared to the models using global descriptors shown in Table 5.3. This means that our semantic analysis considering the visual words described by local descriptors produced video summaries more in accordance to the summaries expected by the users. Between the local descriptors, HUESIFT performed better than the traditional SIFT and got the best results. Meaning that the color information was useful, but the improvement is not significant. The rest of the local descriptors had a similar performance showing the robustness of local descriptors in video summarization.

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Manhattan</th>
<th>Cosine</th>
<th>H. Intersection</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$C_{UA}$</td>
<td>$C_{UE}$</td>
<td>$C_{UA}$</td>
<td>$C_{UE}$</td>
</tr>
<tr>
<td>HueSIFT</td>
<td>0.967</td>
<td>0.033</td>
<td>0.987</td>
<td>0.013</td>
</tr>
<tr>
<td>CSIFT</td>
<td>0.959</td>
<td>0.041</td>
<td>0.981</td>
<td>0.019</td>
</tr>
<tr>
<td>HoG</td>
<td>0.956</td>
<td>0.044</td>
<td>0.985</td>
<td>0.015</td>
</tr>
<tr>
<td>SURF</td>
<td>0.955</td>
<td>0.045</td>
<td>0.982</td>
<td>0.018</td>
</tr>
<tr>
<td>SIFT</td>
<td>0.954</td>
<td>0.046</td>
<td>0.985</td>
<td>0.015</td>
</tr>
</tbody>
</table>

**Tabela 5.2:** (Open Video Data Set) Experiments using temporal segmentation

In Table 5.3, the video segmentation proposed by (de Avila, ao Lopes, da Luz and de Albuquerque Araújo 2011) is used in our method, instead of our proposed temporal segmentation. As we can see, the visual words using local descriptors decreased their performance. This shows that local descriptors can be sensitive to visual video effects, such as dissolves. We can see that VSUMM performed better, but not by much, compared to local descriptors using the Manhattan distance. But when we observe the results obtained with the other distances, there is no significant difference between them. The reason of this, is that VSUMM was proposed using the Manhattan distance. Using other distances, VSUMM obtained better result, but the difference between the performance of VSUMM and the rest of local descriptors decreased to the point of being fairly similar. Another thing that we can observe is that, taking aside VSUMM, the local descriptors had better results than the rest of global descriptors. We must also note that despite decreasing their performance compared to the results with temporal segmentation, the visual words using local descriptors still produce relevant summaries and even better than most of the global descriptors used in our experiments, excluding VSUMM. This helps us to corroborate the importance of temporal segmentation in video summarization when using local descriptors. Additionally, it is noted that HoG and CSIFT produced the best results among the local descriptors without video temporal segmentation.
### Tabela 5.3: (Open Video Data Set) Experiments without temporal segmentation

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Manhattan</th>
<th>Cosine</th>
<th>H. Intersection</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HueSIFT</td>
<td>0.829</td>
<td>0.171</td>
<td>0.896</td>
<td>0.104</td>
</tr>
<tr>
<td>CSIFT</td>
<td>0.858</td>
<td>0.142</td>
<td>0.919</td>
<td>0.081</td>
</tr>
<tr>
<td>HoG</td>
<td>0.869</td>
<td>0.131</td>
<td>0.925</td>
<td>0.075</td>
</tr>
<tr>
<td>SURF</td>
<td>0.808</td>
<td>0.192</td>
<td>0.883</td>
<td>0.117</td>
</tr>
<tr>
<td>SIFT</td>
<td>0.829</td>
<td>0.171</td>
<td>0.909</td>
<td>0.091</td>
</tr>
<tr>
<td>Global</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DT</td>
<td>0.635</td>
<td>0.365</td>
<td>0.706</td>
<td>0.294</td>
</tr>
<tr>
<td>STIMO</td>
<td>0.827</td>
<td>0.173</td>
<td>0.886</td>
<td>0.114</td>
</tr>
<tr>
<td>OV</td>
<td>0.795</td>
<td>0.205</td>
<td>0.831</td>
<td>0.169</td>
</tr>
<tr>
<td>VSUMM</td>
<td><strong>0.901</strong></td>
<td><strong>0.099</strong></td>
<td><strong>0.940</strong></td>
<td><strong>0.060</strong></td>
</tr>
</tbody>
</table>

#### 5.4.2 Results for the Youtube Database

Using the Youtube database, we observe that the evaluation values decreased for both global and local descriptors, although they have the same tendency of the results obtained using the Open Video database. The reason is that the database has another type of videos, such as sports, games and news. In this type of videos, users are more interested in the events rather than the objects. Consequently, a domain-specific summarization is more ideal. Nonetheless, according to Table 5.4, local descriptors still got better results compared to global descriptor VSUMM shown in Table 5.5. Consequently, the overall better performance of local descriptors shows that the proposed semantic analysis and temporal segmentation helps to produce more meaningful summaries.

In Table 5.5, it is also noted that the local descriptors decreased their performance without temporal segmentation, this behavior was also noted in Table 5.3. Compared to VSUMM (Global descriptor), local descriptors still produced good summaries. We can conclude from the two tables that local descriptor’s performance is greatly benefited from temporal segmentation. And also, even without temporal segmentation, local descriptors still provides promising summaries.
### Experimental Results

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Manhattan</th>
<th>Cosine</th>
<th>H. Intersection</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>$CUS_A$</td>
<td>$CUS_E$</td>
<td>$CUS_A$</td>
<td>$CUS_E$</td>
</tr>
<tr>
<td>HueSIFT</td>
<td>0.870</td>
<td>0.130</td>
<td>0.949</td>
<td>0.051</td>
</tr>
<tr>
<td>CSIFT</td>
<td>0.865</td>
<td>0.135</td>
<td>0.932</td>
<td>0.068</td>
</tr>
<tr>
<td>HoG</td>
<td>0.865</td>
<td>0.135</td>
<td>0.943</td>
<td>0.057</td>
</tr>
<tr>
<td>SURF</td>
<td>0.869</td>
<td>0.131</td>
<td>0.948</td>
<td>0.052</td>
</tr>
<tr>
<td>SIFT</td>
<td>0.859</td>
<td>0.141</td>
<td>0.956</td>
<td>0.044</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Manhattan</th>
<th>Cosine</th>
<th>H. Intersection</th>
<th>$\chi^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>$CUS_A$</td>
<td>$CUS_E$</td>
<td>$CUS_A$</td>
<td>$CUS_E$</td>
</tr>
<tr>
<td>HueSIFT</td>
<td>0.749</td>
<td>0.251</td>
<td>0.865</td>
<td>0.135</td>
</tr>
<tr>
<td>CSIFT</td>
<td>0.735</td>
<td>0.265</td>
<td>0.863</td>
<td>0.137</td>
</tr>
<tr>
<td>HoG</td>
<td>0.726</td>
<td>0.274</td>
<td>0.860</td>
<td>0.140</td>
</tr>
<tr>
<td>SURF</td>
<td>0.725</td>
<td>0.275</td>
<td>0.852</td>
<td>0.148</td>
</tr>
<tr>
<td>SIFT</td>
<td>0.746</td>
<td>0.254</td>
<td>0.875</td>
<td>0.125</td>
</tr>
<tr>
<td>Global</td>
<td>$CUS_A$</td>
<td>$CUS_E$</td>
<td>$CUS_A$</td>
<td>$CUS_E$</td>
</tr>
<tr>
<td>VSUMM</td>
<td>0.759</td>
<td>0.241</td>
<td>0.868</td>
<td>0.132</td>
</tr>
</tbody>
</table>

**Tabela 5.4:** (Youtube Data Set) Experiments using temporal segmentation

**Tabela 5.5:** (Youtube Data Set) Experiments without temporal segmentation

### 5.5 Final Results Analysis

According to our experiments, the color information used by the HUESIFT descriptor got the best results in the Open Video dataset and also in the Youtube dataset using the Manhattan distance. Nonetheless, this performance is not drastically superior to others local descriptors that do not use color information, such as SIFT, HoG or SURF. Due to the mixed results of HUESIFT and CSIFT (sometimes better, sometimes worst between the evaluated local descriptors), it is inconclusive whether color leads to more relevant video summaries.

We must outline that the SURF and HoG descriptors also produced promising results. Because the SURF and HoG descriptors are not as computationally expensive to compute as the SIFT, CSIFT and HUESIFT, they are a good option for a faster video summarization considering the semantic information.
Figures 5.1, 5.2, 5.3, 5.4 and 5.5, show the manual summaries of 5 users. These summaries share some similarity but they also have some differences. This shows that the users acknowledge the relevance of some frames and that this notion is shared with other users. There are also frames that are different between the user’s summaries, and this is the subjective appreciation of each user about what they consider to be relevant.

In Figure 5.6, the video summary for the same video using the de Avila, ao Lopes, da Luz and de Albuquerque Araújo (2011) approach is shown. Comparing it to 5.7, we can see that the proposed method using the SURF descriptor was able to recognize most of the user’s selected frames.

We have also made tests using different distance measures at the moment of filtering the video summaries, such as the euclidian, manhattan and cosine distance. But there were no significant change in the final results compared to the measure originally proposed.

Figura 5.1: User Summary Number 1 for video *Drift Ice as a Geologic Agent, segment 08*

### 5.6 Final Considerations

In this chapter, we have defined the different parameters under which, we have produced our experimental results. In order to obtain the values of these parameters, we have used a subset of videos and then experimented with different values until we empirically achieved the values that produced the best results. According to our experiments, local descriptors perform better than global descriptors in video summarization. We can
**Figura 5.2:** User Summary Number 2 for video *Drift Ice as a Geologic Agent, segment 08*

**Figura 5.3:** User Summary Number 3 for video *Drift Ice as a Geologic Agent, segment 08*

**Figura 5.4:** User Summary Number 4 for video *Drift Ice as a Geologic Agent, segment 08*
also observe the importance of temporal video segmentation, and their impact in video summarization using local descriptors. We have applied our proposed method in two different datasets of videos and the results validate our method.
Figure 5.7: Video Summary using SURF descriptor for video *Drift Ice as a Geologic Agent*, segment 08
Capítulo 6

Conclusions and Future Work

Automatic video summarization has been a popular topic of investigation during the last years. Video summarization has been an option to deal with the exponential growth of video information in the internet, security videos, etc. It can also help to reduce the costs of storage and transmission of video content. Furthermore, it can help us catalogue and browse information along large video collections and therefore reduce human effort. Researchers have increased their interest for video summarization, encouraged by the different challenges this topic involves.

In this dissertation, we approached the task of video summarization by considering the semantic information expressed by the video’s visual entities. The proposed method elaborates static video summaries and our core approach is to use temporal video segmentation and visual words obtained by local descriptors. The proposed method has taken advantage of previous techniques in video summarization and segmentation. We show how this approach leads to a successful summarization of generic videos.

Additionally, we evaluate the importance of local descriptors and temporal segmentation in automatic video summarization. We compare our results to other models that use global descriptors and simple video segmentation. According to our experiments, the color information used by some local descriptors did not lead into a greater performance compared to other local descriptors that do not use color information. In addition, video summaries created with our semantic analysis were the most similar to the user’s summaries (ground-truth). Nonetheless, they decrease their performance when no temporal segmentation is used.

We have based our analysis on the results of an evaluation method that allowed us to quantify the quality of the video summaries. Such method allowed us to reduce
the subjectivity of the evaluation task and to quickly compare our results with other approaches.

6.1 Future Work

Following the work described in this dissertation, a number of projects could be taken up:

- Enhance the evaluation method so it can consider the number of correct matching frames and also penalize unwanted frames in the final video summary.

- Evaluate the performance of other global descriptors that exploit information such as texture, shape, movement, etc.

- Evaluate the performance of our approach applied to specific video genres, such as animation, sports, movies, surveillance, etc.

- Study the impact of other techniques to determine the ideal number of segments used in the clustering algorithm.
Bibliografia


