

Lino Miguel Moreira Ferreira

## Methods for Flexible Representation and Coding of 2D and 3D Visual Information

Tese de doutoramento em Engenharia Eletrotécnica e de Computadores, ramo de especialização em Telecomunicações, orientada pelo Professor Doutor Luís Alberto da Silva Cruz e Professor Doutor Pedro António Amado de Assunção e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Abril/2016



UNIVERSIDADE DE COIMBRA





**Universidade de Coimbra**  
**Faculdade de Ciências e Tecnologia**  
**Departamento de Engenharia Eletrotécnica e de**  
**Computadores**

# **Methods for Flexible Representation and Coding of 2D and 3D Visual Information**

**Lino Miguel Moreira Ferreira**

Coimbra

April 2016



# Methods for Flexible Representation and Coding of 2D and 3D Visual Information

Lino Miguel Moreira Ferreira

*Submitted in partial fulfillment of  
the requirements for the degree of  
Doctor of Philosophy*

Universidade de Coimbra  
Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Eletrotécnica e de Computadores

under supervision of

Professor Doutor Luis Alberto da Silva Cruz (advisor)

Professor Doutor Pedro António Amado de Assunção (co-advisor)

Coimbra

April 2016



*To my family, Afonso, Ana Miguel, Vasco and my wonderful wife Raquel.*





## Acknowledgments

This Thesis was a long and sometimes hard journey. It would not have been possible without the financial, technical, moral support, experiences and guidance of many people and institutions. I am indebted to the many people who have made this Thesis possible and, although I would like to express my gratitude to all of them here, it is unfortunately not possible in such limited space. However, some of these people played such an essential part in the conclusion of this Thesis that at least merit to have their names remarked.

First of all, I would like to thank to my scientific supervisors, Professor Doutor Pedro António Amado Assunção and Professor Doutor Luis Alberto da Silva Cruz, by their permanent support, pragmatism and also by their fruitful discussions allowing me to pursue the research work with success.

My gratitude to Instituto de Telecomunicações, in special to Leiria branch, for providing me the physical, material and financial conditions necessary to develop the research activities.

My acknowledgment also to Fundação para a Ciência e a Tecnologia (FCT), that has supported this work with grants SFRH/BD/37510/2007 and R&D Unit UID/EEA/50008/2013, Project 3DVQM co-funded by FEDER-PT2020, FCT/MEC, Portugal.

I would also like to thank all my colleagues of Instituto de Telecomunicações for creating such a friendly and cooperative working environment. In particular, Luís Lucas, Sylvan Marcelino and João Carreira, for their friendly and for their help in solving technical and logistic problems. Also, I want to thank to Pedro Correia, for his friendship and for his collaboration in part of the presented work. My gratitude to Sérgio Silva and Nelson Ferreira for their encouragement, friendship and help in solving technical problems.

I sincerely want to thank to all my family for all support and care that they have given me all this time. Finally, my deep gratefulness to my daughter Ana Miguel, my sons Afonso and Vasco, and my wonderful wife Raquel, for their constant comprehension, support and love.

And last, but not least, I also have to thank all my friends, simply for being there.



## Abstract

Nowadays, there is a great diversity and quantity of image and video content used in multimedia services and applications, which require efficient and flexible management tools for different purposes, such as adaptation, indexing, searching and browsing. However, the existing representation formats are mostly agnostic in regard to the visual content conveyed by the digital signals. As a consequence, the access and processing of the visual information based on user-driven parameters is rather limited and the most efficient solutions for adaptation and matching heterogeneous constraints in communication systems cannot be easily achieved. In this context, the research work carried out in this Thesis is a contribution to advance the current state-of-the-art in regard to methods and models capable of providing different types of additional flexibility in the representation of visual information. In order to understand the current state of these methods, a thorough study of the most relevant aspects of them were presented.

This Thesis begins with a study of the basic concepts used in representation of the visual information either in raw or coded format. A review of visual saliency computation methods for 2D/3D video is presented, where the most relevant methods available in the literature are explained and discussed. A comprehensive study of temporal segmentation and video summarisation methods for 2D/3D is also described. Then an overview of video retargeting methods is presented, addressing different methods and including non-content-aware and content-aware retargeting methods. In addition, an overview of coding schemes that are able to cope with flexible representation of visual content is also described. After a brief review of video coding concepts, the study is mainly focused on scalable and ROI video coding.

The research work developed in the scope of this Thesis addresses several computing methods, able to provide additional flexibility in the representation and coding of visual information. Two methods for computing visual saliency maps for 3D video were firstly proposed. These are based on fusion of four intermediate saliency maps (spatio-temporal, depth and face saliency) followed by a centre-bias weighting function, which models the human tendency to gaze at objects located in the centre of the visual scene. These methods were evaluated and validated with diverse publicly available datasets containing video sequences and the respective ground-truth fixation density maps. The experimental results show that the proposed methods achieve better performance than other state-of-the-art methods available in the literature.

Then, using the saliency maps, representing the visual relevance of different image regions, a spatio-temporal retargeting method based on such salient regions was developed and evaluated. The proposed method is able to resize the original video for any specific display size and when compared against other state-of-the-art methods, the results show that competitive results can be achieved.

Finally, a flexible representation of visual information in the temporal domain was also investigated in the field of video summarisation. A computational framework to obtain compact versions of video sequences (i.e., video summary), according to meaningful criteria was presented, based on a two-step approach: temporal segmentation and the key-frame extraction. The proposed solution is capable of dealing with various video types and formats, using different several criteria to segment the original video sequence and to select the key-frames. Using different performance metrics and publicly available data for comparison, the results demonstrate that the proposed framework outperforms similar state-of-the-art methods.

Overall, the methods investigated in this Thesis and the performance results obtained from extensive simulations, demonstrate that valid contributions to advance the current state-of-the-art were achieved and also good insight for future research.

## **Keywords**

Video content; Visual representation; Visual saliency computational methods; Video summarisation; Video retargeting; ROI coding; Video summary coding.

## Resumo

Atualmente existe uma grande diversidade e quantidade de conteúdos multimídia utilizados em diferentes aplicações que exigem ferramentas de gestão eficientes e flexíveis para diferentes fins, tais como adaptação, indexação e pesquisa. No entanto, os formatos de representação atuais são tipicamente agnósticos em relação ao conteúdo visual contido nos sinais digitais. Conseqüentemente, o acesso e o processamento da informação visual com base em algum tipo de relevância para os utilizadores ficam bastante limitados, e as soluções mais eficientes para adaptação de conteúdos devido a restrições dos sistemas de comunicação heterogêneos podem não ser facilmente alcançadas. Neste contexto, o trabalho de investigação realizado nesta Tese é uma contribuição para aumentar a flexibilidade de representação da informação visual existente nos sinais de vídeo e expandir o atual estado-da-arte relativamente aos métodos associados.

Esta dissertação é iniciada por um estudo dos conceitos básicos utilizados na representação da informação visual codificada e por codificar. Uma revisão dos métodos usados para calcular saliências visuais em vídeo 2D/3D é apresentada, onde uma explicação mais exaustiva foi realizada para os métodos mais relevantes. Apresenta-se também um estudo abrangente dos principais métodos de segmentação temporal e sumarização de vídeo 2D/3D. No seguimento, uma visão geral dos métodos de redimensionamento de vídeo foi apresentado. Adicionalmente, são descritos de forma global os conceitos básicos de codificação de vídeo, incluindo um estudo mais aprofundado da codificação de vídeo escalável e das Regiões de Interesse.

O trabalho de pesquisa desenvolvido no âmbito desta Tese apresenta vários métodos capazes de proporcionar uma flexibilidade adicional aos atuais métodos existentes de representação e codificação da informação visual. Dois métodos para calcular mapas de saliência visual em vídeo 3D foram propostos. Estes métodos, baseiam-se na fusão de quatro mapas de saliência intermédios (espaço-temporal, de profundidade e da saliência face), seguido por uma função de ponderação *centre-bias*, que modela a tendência humana para observar objetos localizados no centro da cena. Estes métodos foram validados e avaliados com mapas de densidade de fixação publicamente disponíveis. Os resultados experimentais demonstram que os métodos propostos obtêm melhor desempenho do que outros descritos na literatura.

Adicionalmente, e utilizando os mapas de saliência visual, que representam a relevância visual das diferentes regiões da imagem, foi desenvolvido e avaliado um método de red-

imensionamento espaço-temporal baseado nessas regiões salientes. O método proposto tem capacidade de redimensionar o vídeo original para o qualquer tamanho de ecrã e quando comparado com outros métodos existentes na literatura, resultados competitivos foram alcançados.

Finalmente, a representação flexível da informação visual no domínio temporal foi investigada no âmbito sumarização de vídeo. Neste caso, foi estudado e proposto uma abordagem nova para obter versões reduzidas de uma sequência de vídeo, de acordo com critérios previamente definidos. Esta abordagem é constituída por duas partes: a segmentação temporal e a extração das tramas-chave. A solução proposta suporta vários formatos de vídeo e pode utilizar diferentes critérios para segmentar o vídeo original e extrair as tramas-chave. Diferentes métricas e vídeos foram utilizadas para avaliar o desempenho da solução proposta.

Os resultados demonstram que a solução apresentada supera outros métodos descritos na literatura para o mesmo fim. No geral, os métodos investigados nesta Tese e os resultados de desempenho obtidos a partir de simulações demonstram a validade do trabalho realizado e são motivadoras para futuras investigações.

## **Palavras-Chave**

Conteúdo de Vídeo; Representação Visual; Métodos para calcular a Saliência Visual; Sumarização de Vídeo; Redimensionamento de Vídeo; Codificação de Regiões de Interesse; Codificação de Sumários de Vídeo.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Main objectives and contributions . . . . .	3
1.3	Outline . . . . .	5
<b>2</b>	<b>Flexible representation of visual information - review</b>	<b>7</b>
2.1	Video formats . . . . .	7
2.1.1	Colour spaces . . . . .	8
2.1.2	2D/3D video formats . . . . .	9
2.2	Visual saliency . . . . .	14
2.2.1	Visual saliency computation methods . . . . .	17
2.2.2	Performance metrics . . . . .	21
2.2.3	Applications . . . . .	24
2.3	Video retargeting . . . . .	25
2.3.1	Non-content-aware video retargeting . . . . .	25
2.3.2	Content-aware video retargeting . . . . .	26
2.3.3	Performance metrics . . . . .	29
2.3.4	Applications . . . . .	30
2.4	Video summarisation . . . . .	32
2.4.1	Shot boundary detection . . . . .	34
2.4.2	Key-frame extraction methods . . . . .	40
2.4.3	Presentation of video summaries . . . . .	48
2.4.4	Performance metrics . . . . .	50
2.4.5	Applications . . . . .	56
2.5	Flexible video coding . . . . .	59
2.5.1	Basic concepts . . . . .	59
2.5.2	Scalable video coding . . . . .	60
2.5.3	ROI based video coding . . . . .	67
2.5.4	Performance metrics . . . . .	70
2.6	Discussion . . . . .	71
2.7	Conclusions . . . . .	73

---

<b>3</b>	<b>Visual saliency computation by feature aggregation</b>	<b>75</b>
3.1	Visual saliency computation methods . . . . .	75
3.1.1	Visual saliency computation using spatio-temporal depth information	76
3.1.2	Improved visual saliency computation method based on face saliency . . . . .	80
3.2	Results and analysis . . . . .	83
3.2.1	Experimental setup and methodology . . . . .	83
3.2.2	Visual saliency computation using spatio-temporal depth information	86
3.2.3	Improved visual saliency computation method based on face saliency	89
3.3	Conclusions . . . . .	91
<b>4</b>	<b>Video retargeting</b>	<b>93</b>
4.1	Spatio-temporal adaptation method based on visual saliency information .	93
4.1.1	Visual saliency map computation . . . . .	94
4.1.2	Determination of the cropping window . . . . .	95
4.1.3	Temporal filtering . . . . .	96
4.1.4	Cropping . . . . .	97
4.2	Hybrid video retargeting method based on visual saliency information . . .	98
4.2.1	Determination of the cropping window . . . . .	99
4.2.2	Spatio-temporal filtering . . . . .	100
4.2.3	Cropping . . . . .	101
4.2.4	Resizing . . . . .	101
4.3	Results and analysis . . . . .	102
4.3.1	Visual comparison . . . . .	103
4.3.2	Temporal consistency-visual comparison . . . . .	108
4.3.3	The influence of temporal consistency on video encoding efficiency .	109
4.4	Conclusions . . . . .	112
<b>5</b>	<b>3D/2D video summarisation</b>	<b>113</b>
5.1	Video shot boundary detection . . . . .	114
5.2	Key-frame extraction methods . . . . .	122
5.2.1	Optimal key-frame extraction method based on minimum reconstruction error . . . . .	123
5.2.2	Fast key-frame extraction method based on MSE and PCA . . . . .	126
5.2.3	3D key-frame extraction based on perceptually relevant depth regions	133



---

5.2.4	3D and 2D key-frame extraction driven by aggregated saliency maps	144
5.3	Conclusions	157
<b>6</b>	<b>Flexible video coding based on spatial and temporal scalability</b>	<b>159</b>
6.1	ROI coding with spatial scalability	160
6.1.1	$QP_{51}$	160
6.1.2	Set-to-Zero	161
6.1.3	Results and analysis	161
6.2	Video summary coding with temporal scalability	167
6.2.1	Dynamic GOP size selection	167
6.2.2	Prediction structure in temporal scalable coding	168
6.2.3	Results and analysis	169
6.3	Conclusions	173
<b>7</b>	<b>Conclusion and future work</b>	<b>175</b>
7.1	Conclusions	175
7.2	Future work	177
<b>A</b>	<b>Published papers</b>	<b>179</b>
A.1	Journal papers	179
A.2	E-letter	179
A.3	Conference papers	179
	<b>References</b>	<b>181</b>



## List of Tables

2.1	Common digital video formats. . . . .	9
3.1	Details of the test sequences. . . . .	84
3.2	PLCC evaluation-proposed and competing methods - Hanhart's FDM data [1]. . . . .	87
3.3	KLD evaluation-proposed and competing methods - Hanhart's FDM data [1]. . . . .	87
3.4	Centre-bias weighting performance with Hanhart's FDM data [1]. . . . .	89
3.5	Performance of the proposed method with face saliency <i>vs</i> Wang's method [2]-Hanhart's FDM data [1]. . . . .	90
3.6	Performance of the proposed method with face saliency <i>vs</i> Wang's method [2]-Fang's FDM data [3]. . . . .	90
3.7	Proposed method without face saliency map-Hanhart's FDM data [1]. . . . .	91
3.8	Proposed method without face saliency map-Fang's FDM data [3]. . . . .	91
4.1	Details of the test sequences used in the experiments. . . . .	103
5.1	Details of the test sequences used in the experiments. . . . .	120
5.2	Results of the 3DSB detection. . . . .	122
5.3	Key-frame summary notation. . . . .	123
5.4	Details of the test sequences. . . . .	130
5.5	Performance of the proposed method <i>vs</i> Zhu Li's [4] method for <i>Foreman</i> sequence. . . . .	131
5.6	Performance of the proposed method <i>vs</i> Zhu Li's [4] method for <i>Mother&amp;Daughter</i> sequence. . . . .	131
5.7	Computational complexity of the <i>Foreman</i> sequence. . . . .	132
5.8	Computational complexity of the <i>Mother&amp;Daughter</i> sequence. . . . .	132
5.9	Key-frames of the <i>Foreman</i> sequence. . . . .	132
5.10	Key-frames of the <i>Mother&amp;Daughter</i> sequence. . . . .	133
5.11	Characterisation of the test sequences used in the experiments. . . . .	140
5.12	Key-frame ratio. . . . .	140
5.13	3D summarisation - $\Delta_{SRD}$ comparison: proposed <i>vs</i> UnS and Clu methods. . . . .	142
5.14	3D summarisation - $\Delta_{Fm}$ comparison: proposed <i>vs</i> UnS and Clu methods. . . . .	142
5.15	SRD measure results. . . . .	143
5.16	Fidelity measure results. . . . .	143

5.17	Details of the test sequences. . . . .	148
5.18	Aggregated saliency maps for 3D video: performance comparison. . . . .	151
5.19	2D key-frame extraction: proposed <i>vs</i> non-saliency based methods. . . . .	151
5.20	2D key-frame extraction: proposed <i>vs</i> saliency based methods. . . . .	152
5.21	Key-frames extracted from sequence video <i>Hurricane Force - A Coastal Perspective, segment 03</i> . . . . .	154
5.22	3D key-frame extraction: proposed <i>vs</i> UnS and AtC methods. . . . .	155
5.23	Performance of the proposed key-frame extraction method for 3D video. . .	156
6.1	Processing time of the intra coding mode. . . . .	164
6.2	Processing time of the inter coding mode. . . . .	165

## List of Figures

2.1	Stereoscopic pair (a,b) and corresponding difference (c) of the <i>Ballet</i> sequence. . . . .	10
2.2	Example of MVV format for <i>Ballet</i> sequence. . . . .	11
2.3	Representation of the texture (a) and associated depth-map (b) of the <i>Ballet</i> sequence. . . . .	11
2.4	Example of MVD format for <i>Ballet</i> sequence. . . . .	12
2.5	3D model example without (a) and with colour (b). . . . .	13
2.6	Different focal planes extracted form a 3D holoscopic image. . . . .	13
2.7	A prototypical example of bottom-up attention. . . . .	15
2.8	Original image of a dog playing with ball (a) and its saliency map (b) (the white regions indicate the more salient parts of image). . . . .	16
2.9	Computed saliency map of the <i>News report</i> sequence(a) $\tau = 10$ (b) $\tau = 100$ (c) $\tau = 200$ (d). . . . .	24
2.10	Examples of image retargeting for devices with different display resolutions. . . . .	26
2.11	Non-content-aware cropping and scaling methods. . . . .	27
2.12	A conceptual framework for key-frame summarisation. . . . .	34
2.13	A generic diagram of SBD framework. . . . .	36
2.14	Point distance of the frame #38 and #39 of the <i>Batter</i> sequence. Grey values means the point distance from (0,0,0) [5]. . . . .	37
2.15	a) UnS-method: uniform sampling at equal intervals. b) PoS-method: selecting the first frame of each video shot. . . . .	41
2.16	Video synopsis proposed [6]. . . . .	49
2.17	(a) 3D-Ring interface, (b) 3D-Globe interface and (c) 2D grid presentation (figure based on [7]). . . . .	50
2.18	Single image key-frame presentation method [8]. . . . .	50
2.19	Scalability types. . . . .	61
2.20	Simplified SVC encoder architecture [9]. . . . .	62
2.21	Temporal scalable architecture of SVC [10]. . . . .	63
2.22	Multi-layer structure with inter-layer prediction [11]. . . . .	64
3.1	Functional diagram of the visual saliency computation method. . . . .	77
3.2	Functional diagram of the improved visual saliency computation method. . . . .	81
3.3	Example face saliency map of the <i>News report</i> sequence. . . . .	82

3.4	First frame of the original video, Fang’s [3] and Hanhart’s [1] FDM data respectively. . . . .	85
3.5	Visual saliency - Prop. and competing methods for the frame 120 of the <i>Boxers</i> sequence. . . . .	88
3.6	Visual saliency - proposed method with and without face saliency maps <i>vs</i> competing methods for the frame 120. . . . .	92
4.1	Functional diagram of the video retargeting method. . . . .	94
4.2	Functional diagram of visual saliency computation method. . . . .	95
4.3	Red box is the cropping window ( $1280 \times 720$ ) of the <i>Jockey</i> sequence. . . . .	96
4.4	Temporal evolution of the upper-left corner of the cropping window - <i>Jockey</i> video sequence. . . . .	97
4.5	Functional diagram of the multi-operator video retargeting method. . . . .	98
4.6	Red box is the cropping window with 70% of the total energy - <i>Bosphorus</i> video sequence. . . . .	101
4.7	Result of the application, (a) cropping operation to $2517 \times 1154$ , (b) down-sizing operation to $1280 \times 720$ - <i>Bosphorus</i> sequence. . . . .	102
4.8	Visual comparison of retargeted methods of the <i>Jockey</i> sequence. . . . .	105
4.9	Visual comparison of retargeted methods of the <i>Bosphorus</i> sequence. . . . .	106
4.10	Visual comparison of retargeted methods of the <i>HoneyBee</i> sequence. . . . .	107
4.11	(a)-(d)Four consecutive frames of <i>Jockey</i> sequence. Retargeted (e)-(h) without median filter and (i)-(l) with median filter. . . . .	108
4.12	(a)-(d)Four consecutive frames of <i>Bosphorus</i> sequence. Retargeted (e)-(h) without median filter and (i)-(l) with median filter. . . . .	109
4.13	R-D of the <i>Bosphorus</i> sequence. . . . .	110
4.14	R-D of the <i>Jockey</i> sequence. . . . .	111
4.15	R-D of the <i>HoneyBee</i> sequence. . . . .	111
5.1	3DSB algorithm architecture. . . . .	116
5.2	Feature vectors of all 3DSB candidates frames. . . . .	117
5.3	Initial centroids of the two clusters. . . . .	118
5.4	Selection of the 3DSB frames. . . . .	119
5.5	<i>Summer in Heidelberg</i> sequence: 12 frames corresponding to dissolve smooth transition (a)-(f) and sharp transition (g)-(l). . . . .	121
5.6	<i>Oldtimers</i> sequence: 12 frames (3 from each shot) corresponding to 2 sharp transitions. . . . .	121
5.7	Frame-by-frame distortion of the <i>Foreman</i> sequence. . . . .	128

5.8	Functional diagram of the 3D key-frame extraction method based on perceptually relevant depth regions. . . . .	134
5.9	Steps for the calculation depth relevance region of the <i>Pantomime</i> sequence.	136
5.10	Calculation the depth relevance region of the <i>Ballet</i> sequence. . . . .	137
5.11	Framework for 2D/3D key-frame extraction. . . . .	147
6.1	$Qp_{51}$ functional diagram. . . . .	161
6.2	<i>Set-to-Zero</i> functional diagram. . . . .	162
6.3	ROI1 (left) and ROI2 (right) of the <i>Mobile</i> sequence outlined in red. . . . .	163
6.4	R-D performance for the intra coding case. . . . .	164
6.5	R-D performance for the inter coding case. . . . .	166
6.6	Example of a prediction structure resulting from dynamic GOP allocation.	169
6.7	R-D performance of the temporal base layer (T0) of the <i>Soccer</i> sequence. .	170
6.8	R-D performance of the temporal base layer (T0) of the <i>Foreman</i> sequence.	171
6.9	R-D of the full rate of the <i>Soccer</i> sequence. . . . .	172
6.10	R-D of the full rate of the <i>Foreman</i> sequence. . . . .	173





# List of Abbreviations

3DSB	3D Shot Boundaries
AIM	Attention based on Information Maximisation
ASB	Abrupt Shot Boundary
AtC	Attention Curve
AVC	Advanced Video Coding
AVDP	Audio-Visual Description Profile
CABAC	Context-Adaptive Binary Arithmetic Coding
CBP	Coded Block Pattern
CCTV	Closed Circuit Television
CGS	Coarse Grain Scalability
CIF	Common Intermediate Format
CUS	Comparison of User Summaries
DCT	Discrete Cosine Transform
DIBR	Depth Image-Based Rendering
DOF	Degrees of Freedom
DOG	Difference of Gaussian
DPB	Decoded Picture Buffer
DSM	Depth Sense Metric
DT	Delaunay Triangulation
FDM	Fixation Density Map
FGS	Fine Grain Scalability
FMO	Flexible Macroblocks Ordering
GOP	Group of Pictures
GSB	Gradual Shot Boundary
HDTV	High Definition TV
HEVC	High Efficiency Video Coding

HVS	Human Visual System
ICA	Independent Components Analysis
JCT-VC	Joint Collaborative Team on Video Coding
JM	Joint Model
JSVM	Joint Scalable Video Model
KLD	Kullback-Leibler Divergence
MB	Macro-Block
MF	Matrix Factorisation
MGS	Medium Grain Scalability
MOS	Mean Opinion Scores
MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
MV	Motion Vectors
MVC	Multiview Video Coding
MVD	Multiview Video plus Depth
NAL	Network Abstraction Layer
NSS	Normalized Scanpath Saliency
PCA	Principal Component Analysis
PDF	Probability Density Function
PDM	Probability Distribution Modelling
PLCC	Pearson Linear Correlation Coefficient
PoS	Position Sampling
PQFT	Phase Spectrum of Quaternion Fourier Transform
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Intermediate Format
QP	Quantisation Parameter
R-D	Rate-Distortion

RMSE	Root Mean Square Error
ROI	Region-of-Interest
SBD	Shot Boundary Detection
SCC	Short Colour Change
SNR	Signal-to-Noise Ratio
SRD	Shot Reconstruction Degree
SSIM	Structural Similarity Index Measure
STIMO	Till and MOving Video Storyboard
SVC	Scalable Video Coding
SVM	Support Vector Machine
TIS	Temporal Image Signature
TSR	Temporal Spectral Residual
UHD	Ultra High Definition
UnS	Uniform Sampling
VCEG	Visual Coding Experts Group
VF	Visual Feature
VSUMM	Video SUMMarisation



# Introduction

---

Nowadays, the technological field of multimedia communications provides support for many applications and services, which are continuously evolving in terms of new features offered to users as well as requirements imposed by the underlying infrastructure, such as equipment, user devices and networks. Current usage environments are also quite diverse, thus seamless access to multimedia content requires different types of adaptation functions along the communication chains to ensure that user parameters, device characteristics and networking resources match together to achieve acceptable Quality of Experience (QoE).

Beyond content adaptation to match devices and networks a further level of content flexibility has been emerging in the last few years in order to allow user access and processing at different levels of information contained in digital multimedia signals. For such purpose, and taking into account that visual information is the most demanding type of content used in multimedia applications and services, due to the huge amount of data involved, flexible representation formats and processing methods specifically targeted to deal with video signals have been under research. In this context, flexible representation of visual information allows either access or processing of only part of the information contained in video signals, selected according to some relevant user-driven criteria.

Particularly relevant examples where flexible representation of visual information is required, include identification and representation of image/video regions attracting different levels of user attention, concise representation of long video sequences using a compact set of representative frames and consistent sequences of sub-images containing the most relevant content from very high resolution image/video. To benefit from such diverse type of information, that is intrinsically embedded in video signals, but not explicitly accessible, it is necessary to devise efficient computational methods capable of identifying and selecting the relevant data and then representing the corresponding information either in raw or coded formats. The research work carried out in this Thesis lies in this context and specifically contributes for advances in visual attention models, video summarisation and retargeting, all of them providing additional levels of content representation flexibility.

## 1.1 Motivation

The recent advances in video technology associated with the increased availability of bandwidth and the incredible popularity of the social networks with billions of users around the world, make digital images/video the most important component of multimedia information. Furthermore, compressed video is also rapidly evolving due the increasing challenging requirements of applications and services, which span over several area such as entertainment, surveillance, medical application, education, etc.

In the last years, new functionalities have been implemented in video applications due to users demand, who are always seeking for new viewing experiences, more interactive and immersive, such as provided by 3D multimedia. Thus, 3D multimedia systems have received increasing attention from the industry and research community due to its higher capability of providing immersive experiences to users. Such immersive experiences are mostly a result of depth sensation provided by the 3D systems, as this is the extra perceptual dimension that makes the difference to classic 2D video. Although inclusion of the depth information in video content is not a recent innovation, the interest in this type of content has been increasing during the last years and the increased availability of 3D video content is also contributing to advances in related technology regarding acquisition, coding and transmission. The recent developments in Ultra High Definition Television (UHDTV) (4K,8K) are also contributing for new requirements in adaptation engines and efficient computational methods to deal with the huge amount of data associated with these formats.

To extract, process and efficiently represent different types of content information in video signals, several methods can be used according to the relevant specific objective. For instance, to find a short and concise representation of a long video sequence, a summary can be computed, comprising few key-frames of the most important content. Basically, a video summary is a short version of a full-length video that preserves the essential visual and semantic information of the original unabridged content. In contrast to summarisation of 2D video, which has been thoroughly investigated in the recent past, 3D video summarisation is still relatively unexplored. Video summarisation methods must preferably be based on the high level semantic contents like people, objects, events and action, but in general identification and extraction at such a content level is only possible in constrained environments. The most common and generic summarisation methods use low level features to select the key-frames from the video. To extend the existing methods,

by including user-driven criteria in video summarisation, visual saliency may be used to increase the perceptual relevance.

In spite of the growing capabilities of storage, transmission systems and processing power, video coding requirements also increase to cope with the huge amount of data produced by new video applications. Until now, several video coding schemes have been proposed which can operate at fixed set points of spatio-temporal resolution and quality. However, scalable video coding is the only one that allows partial decoding of compressed streams with different resolution, either spatial or temporal and/or quality. This functionality can be implemented by dropping part of the video bitstream in order to adapt its characteristics to different needs or preferences of the users as well as terminal capabilities (display resolution, processing power or battery power) or network conditions. Due to flexibility and efficiency of scalable coding in comparison to simulcast, scalable video coding has been attracting wide attention in research community in the recent past with 2D video and more recently with 3D video.

In this context, efficient combinations of spatio-temporal representation methods and scalable video coding techniques are necessary to provide increased content access flexibility. A possible approach for such problem is to generate spatial Region-of-Interest (ROI)s and temporal summaries of 2D and 3D video based on visual saliency maps. By embedding these new features in scalable codecs, the current systems can be extended to new scalable dimensions.

## 1.2 Main objectives and contributions

The main objective of this Thesis is to investigate methods capable of computing new flexible representations of visual information and, at same time, devising efficient coding schemes capable of coping with such information in some useful manner. The methods investigated in this work take into account the relevant visual information in video signals by considering user-driven content features, that can expand usage environments and better match with user preferences. This research work is mainly focused on visual saliency computation methods, video retargeting and video summarisation methods.

The most relevant contributions of this Thesis, related to these objectives, can be summarised as follows:

**Visual saliency computation methods** - Research and development of two methods for computing a spatio-temporal and depth saliency maps for 3D video. These methods are based on fusion of intermediate saliency maps which are obtained from visual features extracted from different domains (spatio-temporal, depth and face saliency). The combination of these intermediate saliency maps with a centre-bias function achieved good results. The proposed methods are not restricted to 3D video and are also applicable to 3D image, 2D image/video. Part of this work was published in J1, E1, C1, and C2

**Video retargeting** - Research, design and development of a method which uses saliency information to resize or crop the original video to smaller resolutions. The retargeting method locates a sub-region of the original video with the pre-defined resolution, which contains the most salient content, i.e., visually relevant regions of the original content. The proposed solution includes a filter to ensure a high level of temporal consistency which stabilizes the position of the cropped or resized area. This work and part of the experiments were published in E1.

**2D/3D video summarisation** - Research and implementation of a framework for automatic selection of the most important frames of a sequence, ensuring that the most relevant visual information of the original video is preserved. This framework is composed of two major processing stages. In the first stage, the video is divided into temporal segments comprising frames with similar content. In the second stage, a set of key-frames is chosen for each temporal segment according to some relevance criteria. Since the proposed framework is compatible with the use of various video types and formats, perceptually meaningful criteria can be used to segment original video and to select the key-frames, e.g., visual saliency. A new key-frame extraction method for 2D/3D video based on aggregated saliency maps is introduced as well as two other summarisation solutions. This work and part of the experiments were published and presented in J1, C3, C4, C5, C8 and C9.

**Flexible video coding** - Research, design and implementation of flexible video coding methods based on spatial and temporal scalability for encoding ROI and video summaries. In the case of video summary coding, the proposed approach is to encode the video summary as the base layer of a scalable bitstream. Using this type of coded representation only base layer needs to decode to access the video summary of a whole video sequence, without the need of fully decoding it. The method proposed for ROI coding enables the



use of differentiated quality and protection against transmission errors for the ROI and non-ROI regions. The results of these explorations were published in C6, C7 and C10.

The contributions presented in this Thesis have been published in several conferences proceedings and journals. The complete list of publications is available at end of this Thesis before the references.

## 1.3 Outline

This Thesis is organized in seven chapters and one appendix. Most chapters start with a short introduction, which is followed by the detailed description of their main content. The ensuing sections include performance evaluation and discussion of results. A brief conclusion finalizes each chapter. Every chapter and appendix is shortly summarised below.

Chapter 2 is intended to provide a general overview of digital representation of images/video and presents a review on methods and formats that enable flexible content representation. Those with particular relevance for this Thesis are visual saliency computation, summarisation and retargeting methods, as well as scalable video coding. This chapter is concluded with a discussion about the state-of-the-art methods, from which several research points were identified.

In Chapter 3, two visual saliency computation methods for 3D video are proposed and evaluated using publicly available fixation density reference datasets. This methods are based on fusion of features maps which contain information from spatio-temporal, depth dimensions and face detection.

A spatio-temporal method for retargeting, based on visual saliency information for UHD video is proposed in Chapter 4. The solution includes temporal filtering to remove the jitter and improve temporal consistency. Moreover, a comparison study with other state-of-the-art methods is presented.

In Chapter 5, three summarisation methods are presented which can be used to construct a compact version of an entire video sequence, while at same time preserving the most relevant visual information. The effectiveness of the proposed methods is then evaluated by comparing them with different summarisation methods available in the literature.

In Chapter 6, two methods for coding ROIs and video summaries with H.264/MPEG-4

SVC encoder are presented and their performance is evaluated. The ROIs and video summaries are obtained by pre-processing using the methods described in previous chapters.

Finally, Chapter 7 concludes the Thesis and provides some possible future research directions associated with this work.

Appendix A includes all papers published in journals and conferences as a result of the work done within the scope of this Thesis.

# Flexible representation of visual information - review

---

This chapter presents a review of video formats and methods for computing relevant visual information based on user-driven criteria, with the objective of extending representation of visual information beyond simply pixels and frames. The aim of creating different types of flexible representation formats is also to enable extraction or visualisation of useful information beyond straightforward content-agnostic and application-independent representation formats. The chapter starts with a brief description of digital representation of images/video and then presents a review on visual saliency computation, summarisation and retargeting methods and scalable video coding, as these are considered as providing flexible representation of the visual information contained in video signals. Finally, the chapter presents a discussion about the state-of-the-art methods, from which several research questions are identified.

## 2.1 Video formats

This section provides a brief description of the structure and characteristics of digital video, addressing colour spaces, sampling formats and video formats, which comprise the data signals used to compute different types of visual information with meaningful relevance for users.

A video signal is a representation of a visual scene either from real-world or synthesized, projected onto a 2D plane. A visual scene is composed of number of dynamic objects, each one with their intrinsic characteristics as shape, motion texture, illumination and depth. The video signal representing the visual information, which is obtained by sampling the color in the spatial and temporal dimensions. Both the spatial and temporal resolution determine the accuracy of the visual representation and have impact on the perceived quality and on amount of data.

### 2.1.1 Colour spaces

Colour space models are used in visual content acquisition and representation to describe the components of the visible light using a reduced numerical range. The RGB (red/green/blue) and YCrCb (luminance/red chrominance/blue chrominance) are the two widely used colour spaces of interest in this context. In RGB, each pixel is represented by three samples, each one corresponding to the level of red, green and blue, respectively. Any other colour can be obtained by the combination of these three components. However, the RGB colour space is not the most efficient model for colour representation, because the Human Visual System (HVS) is less sensitive to colour than luminance which allows lower resolution in colour and less data.

Using the YCbCr colour space it is possible to represent colour images more efficiently than RGB, by separating the luminance from the colour information and using different resolutions for each component. The colour information is represented by two chrominance components (Cb,Cr) and which the luminance is the other component. This is used as in current video and image compression standards such as the HEVC, H.264/MPEG-4 AVC and JPEG. In the YCbCr colour space, Y is the luminance component i.e., a monochrome version of the colour image and it can be computed as a weighted average of R, G and B, as given by Equation (2.1).

$$Y = k_r R + k_g G + k_b B \quad (2.1)$$

where  $k_r$ ,  $k_g$  and  $k_b$  are weighting factors and  $k_r + k_g + k_b = 1$ . Equations (2.2) and (2.3) are used to convert RGB into the YCbCr and vice versa.

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B & R &= Y + 1.402Cr \\ Cb &= 0.564(B - Y) & G &= Y - 0.344Cb - 0.714Cr \\ Cr &= 0.713(R - Y) & B &= Y + 1.772Cb \end{aligned} \quad (2.2) \quad (2.3)$$

As mentioned above, an advantage of YCbCr over RGB is to reduce the amount of data required to represent the chrominance components without having a noticeable effect on visual quality. For a normal observer, there is no evident difference between an RGB image and an YCbCr image with lower chrominance resolution.

Three chroma sub-sampling patterns are normally used for YCbCr, 4:4:4, 4:2:2 and 4:2:0.

In the case of 4:4:4, the three components YCrCb have the same resolution. In the 4:2:2 chroma sub-sampling pattern, the Cb and Cr have the same vertical resolution but half the horizontal resolution, i.e., for every 4 luminance samples in the horizontal direction there are 2 Cb and 2 Cr samples. The last sub-sampling format 4:2:0, reduces Cb and Cr to half of the horizontal and vertical resolution of Y. This chroma sub-sampling format is the common format used for broadcast, internet streaming and content delivery.

### 2.1.2 2D/3D video formats

Table 2.1 shows the sampling parameters (spatial and temporal) for the most common video formats, which are 2D formats normally used for storage and transmission as compressed streams. The choice of the most adequate format is obviously dependent on the target application or service. Tailoring these pure signal-based representations, either in raw or compressed formats, to different usage environments and requirements, is one of the objectives of video summarisation and retargeting methods described in Sections 2.4 and 2.3. Flexible video coding through scalability is another option to consider when different content versions need to be extracted from a single representation (Section 2.5).

Table 2.1: Common digital video formats.

Format	Resolution (H. $\times$ V.)	Temporal sampling [fps]	Raw bit rate (30fps, 8/10 bits) [Mbps]
UHDTV [12]	Lum:7680 $\times$ 4320 Chrom:3840 $\times$ 2160	24,25,30,50,60,120	14930 (10bits)
HDTV [13]	Lum:1920 $\times$ 1080 Chrom:960 $\times$ 540	24,25,30,50,60	933.1 (10bits)
SDTV [14]	Lum:720 $\times$ 576 Chrom:360 $\times$ 288	25, 30	149.3 (8bits)
CIF [14]	Lum:352 $\times$ 288 Chrom:176 $\times$ 144	10-30	36.5 (8bits)
QCIF [14]	Lum:176 $\times$ 144 Chrom:88 $\times$ 72	5-30	9.1 (8bits)

### 3D video

3D video refers to a representation format which differs from 2D video by the implicit or explicit inclusion of depth information about the visual scene. This depth information can be conveyed either implicitly via two or more views of the scene (e.g., left and right

views) or explicitly through depth maps that accompany single or multiple 2D video views. Given the use of 3D video in this Thesis, the most common 3D formats available in the literature are shortly described in the following.

**Stereoscopic video** - it is composed of two slightly shifted views of the same scene, where one corresponds to what would be observed by the left eye and the other by the right eye of a human observer. Figure 2.1 illustrates a stereo image pair and the difference image, where the horizontal disparities can be observed 2.1c.

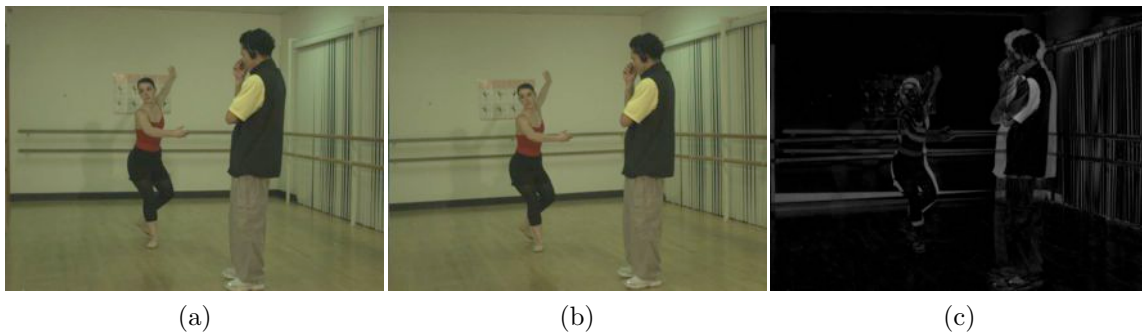


Figure 2.1: Stereoscopic pair (a,b) and corresponding difference (c) of the *Ballet* sequence.

**Multiview Video (MVV)** - it is composed of more than two views slightly shifted in the vertical and/or horizontal position. Typically, MVV acquisition is done using an array of synchronized cameras with some spatial arrangement, which capture the visual scene from different viewpoints. This video format can be used with freeview display systems where a view specified by the viewer is interpolated from the multiple views and presented in a 2D display [15, 16]. Alternatively the MVV format can be used with autostereoscopic displays with or without head tracking, which render a denser set of views that are displayed through lenticular and parallax barriers. With this type of display, viewers are able to see the portrayed scene from different viewpoints. Figure 2.2 shows an example MVV format.

**Video-plus-Depth (V+D)** - it is composed of a video signal (texture) and respective depth map, where each value represents the distance of the object to the camera for the corresponding pixel position. Typically, the depth information is quantized with 8 bits, with the closest point represented with value 255 and the most distant point with 0. Additional views representing the same scene captured from slightly shifted perspectives can



Figure 2.2: Example of MVV format for *Ballet* sequence.

be generated from the original video-plus-depth information by 3D warping transformations [17]. This format has inherent backward compatibility with 2D video systems and higher compression efficiency when compared to stereoscopic video. Figure 2.3 presents an example of the texture and associated depth-maps for the *Ballet* sequence.



Figure 2.3: Representation of the texture (a) and associated depth-map (b) of the *Ballet* sequence.

**Multiview Video-plus-Depth (MVD)** - it is composed of multiple views and corresponding depth maps from which other virtual view can be computed [17]. For example, if a multiview display requires nine video views (V1-V9) simultaneously, delivering nine views directly to the display, would be very costly in terms of bandwidth. Using MVD, it is possible to transmit only three original views (V1, V5 and V9) for instance, along with corresponding depth maps (D1, D5 and D9), and then synthesize the remaining views (V3, V4, V6, V7 and V8) at the decoder side using Depth-Image-Based Rendering (DIBR) techniques [18, 19]. The savings in bandwidth are obvious, at the expense of increased computational capacity for rendering the full set of views in the display. Several emerging applications such as free viewpoint video can use MVD format. An example of

MVD format is presented in Figure 2.4, where the first row is the representation of the texture and second row is the associated depth-map for 3 views.



Figure 2.4: Example of MVD format for *Ballet* sequence.

**3D computer graphics** - this is a geometry-based representation, where the scene is described by a set of connected 3D points (or vertices), with associated texture/colour mapped onto them. The data content of this format can be organized into geometry, appearance and scene information [20]. The geometry of a 3D model includes the position of 3D points (vertices) and polygons (faces) that are constructed by joining these vertices. In the most cases the polygons are plane surfaces defined by three vertices. The appearance is an optional attribute which associates some properties (colour, texture coordinates) to the geometry data. Finally, the scene information includes the layout of a 3D scene with reference to the camera (or view), the light source and description of other 3D models if they are present in the scene. 3D computer graphics can provide better immersive and interactive experience than conventional 2D video, since the user has more freedom to interact with the content and get the feeling of “being there”. Figure 2.5 shows two 3D models (with different views) from the same person without (2.5a) and with colour (2.5b).

**Holographic video** - it is composed of a very large of the number of views captured simultaneously. This multiple view acquisition process can be interpreted as a partial sampling of the plenoptic function [21]. This format, also known as light field, represents not only spatial or temporal information but also angular information of the imaged scene,



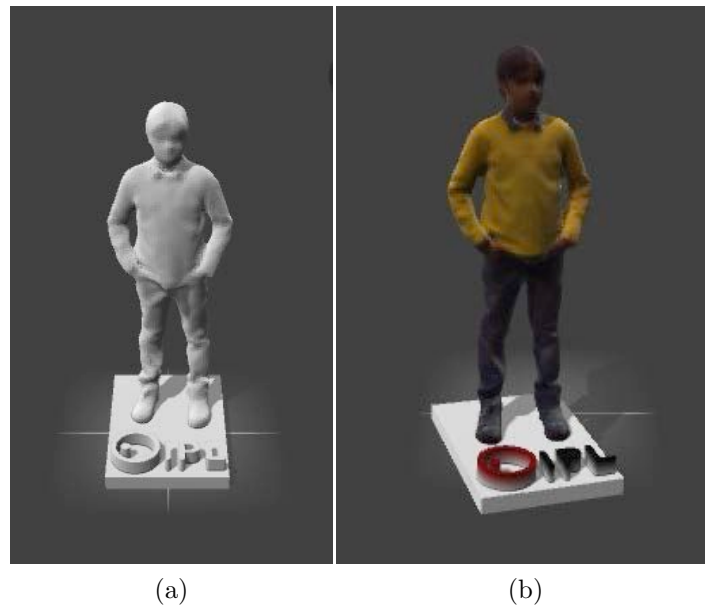


Figure 2.5: 3D model example without (a) and with colour (b).

i.e., captures a segment of the whole observable scene. In practice a 3D holographic image is captured by a normal image sensor placed behind an array of uniformly spaced semi-spherical micro-lenses. Each micro-lens works as an individual low resolution camera that captures the scene from an angle (viewpoint) slightly different from that of its neighbours. This format allows different focal planes of the visual scene (see Figure 2.6).

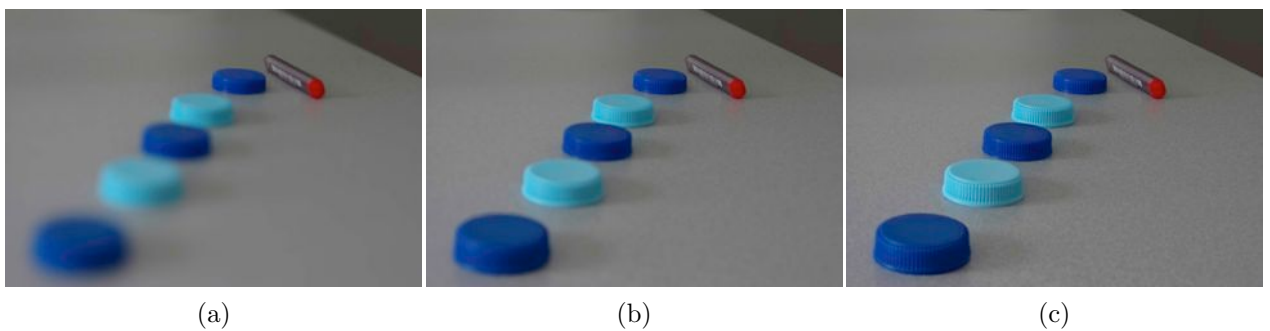


Figure 2.6: Different focal planes extracted from a 3D holographic image.

## 2.2 Visual saliency

Visual saliency is defined as the regions which attract more human visual attention in images and video and may be identified by eye-tracking devices or computational models capable of reproducing the human visual attention. The output of computational models consists of a visual saliency map which represents the level of attention that users tend to draw towards the different areas/objects in the scene. This section presents a review of visual saliency computation methods for computing digital representations (i.e., saliency maps) of the human visual attention when watching 2D/3D video. Some relevant multimedia applications are also discussed.

Since the human visual perception is a result of diverse brain processing functions, which do not assign the same importance to the whole visible area where the scene is happening, it is important to devise computational methods capable of differentiating which data is more relevant in digital images and video. Such methods are able to choose or prioritise the visual representation data according to the selective attention process of human observers. Several studies demonstrated that only a few aspects of the visual information are processed at higher level by the visual system. Hence, visual attention can be seen as the process for reducing the amount of visual information that can be processed by the human brain. As Itti [22], other researchers demonstrated some relations between the mechanisms used by humans to control where they deploy spatial or focal attention and a view scene and the scene content. Therefore, visual attention controls the information that is selected and ensures that such information is relevant to further behavioral decisions.

### Visual attention mechanisms

A visual saliency characterizes some regions or objects of a scene which appear to viewers to stand out from the neighbouring areas, grabbing their attention. According to this visual saliency definition, several questions immediately emerge: “How some regions stand out more than others?” and “How is this process related with the human visual system?” The answer to these questions, was first addressed by Treisman and Gelade [23] in “Feature Integration Theory”, where they alleged which visual features (such as colour, orientation, size, and spatial frequency) are important and how they are combined to direct human attention over pop-out.

Two explanatory mechanisms of visual attention have been proposed in the literature bottom-up and top-down [24], [25] respectively. The first mechanism relates to involuntary, fast, automatic, and unconscious aspects of vision. It is mostly driven by the properties of the visual objects themselves. An example of bottom-up attention, is looking at an image with only one horizontal bar among several vertical bars. The attention is immediately focused to the horizontal bar, as shown in Figure 2.7.

The second mechanism, also known as the task dependent attention, relates to voluntary and conscious aspects of vision, i.e., it is under the control of the person who is attending. These type of mechanisms are used to search for a particular target or to potential targets. For example, a given instruction such as look for the horizontal bar in Figure 2.7.

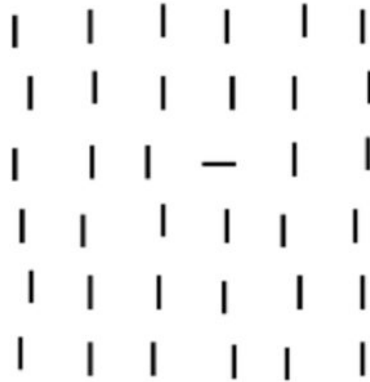


Figure 2.7: A prototypical example of bottom-up attention.

These two mechanisms interact with each other and influence the human visual behaviour [26]. Models of visual attention are designed to produce or predict the saliency maps, which represent the location and level of visual interest of each ROI or frame in the video. The saliency maps are represented as a grayscale image, where the white regions represent higher salient regions and black regions are the less salient ones. Figure 2.8 shows the original image and the respective saliency map for the dog image.

This Thesis deals with the bottom-up visual attention models since they only depend on video/image content, have lower computation complexity and thus are faster than top-down models.

### Eye movement and visual attention

The eye movements convey important information about the cognitive processes of reading, visual search and scene perception. For that, they often are used to detect shifts

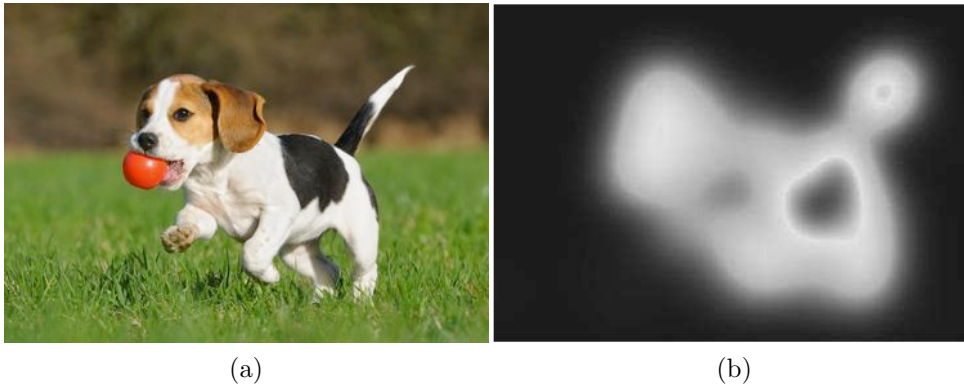


Figure 2.8: Original image of a dog playing with ball (a) and its saliency map (b) (the white regions indicate the more salient parts of image).

of attention. For example, in visual search and scene perception, when the stimulus is more cluttered, fixations become longer and saccades (a quick, simultaneous movement of both eyes between two phases of fixation in the same direction) become shorter [27]. From numerous type of eye movements, the saccadic eye movements are the most studied. Yarbus *et al.* [28] was the first to verify a relationship between saccadic eye movements and visual attention. More recent studies considered that visual attention anticipates an eye movement at the same scene location, i.e., the attention mechanisms inspect several targets and focus in the most important, to which the eyes are then shifted. This leads to the relevant question of “How the human visual attention mechanism selects the scene regions to be attended?” The answer was obtained by Yarbus’s study based on several subject questions, but this type of methodology influence the eye movements (top-down mechanism). However, as mentioned above, bottom-up mechanisms are independent of the observer’s knowledge, motivations and viewing tasks, thus more adequate to compute salient regions from the scene data itself.

The advance of visual attention computational models is dependent on the existence of tools for validation and benchmarking, such as ground truth databases. Many works rely on the Fixation Density Maps (FDM) computed from the eye-tracking experiments. The transformation of eye-tracking raw data into FDM follows several steps: After gathering the raw data from the eye-tracking experiments, saccades are identified and fixation locations are determined. The FDM is obtained by convolving the fixation locations with a Gaussian whose the size is determined by a combination of the mean eye tracking error and the size of the human fovea. Publicly available FDM databases can be found at [1, 3]. These were also used to validate and evaluate the methods investigated in this Thesis.

### 2.2.1 Visual saliency computation methods

In this section, the foundations of visual saliency computation methods for 2D and 3D video are presented, taking into account the background that is necessary for this work. Most visual computational methods for 2D and 3D video proposed in the literature are based on combination of different features extracted from the spatial, temporal and depth-related information.

#### Saliency detection using spatial information

Detecting visual saliency from spatial information (still image) has been studied based on several different approaches. The following three well-known bottom-up approaches of visual attention models were selected as the most relevant to be described. The cognitive model by Itti *et al.* [22], spectral analysis model by Hou and Zang [29] and information theoretic by Bruce and Tsotsos [30].

**Itti’s model [22]** - this model is often used as the basis for other models and it is also employed as a benchmark for comparison. Among others, one the reasons for this success are the quality of documentation and the availability of source code. The model is based on the human vision characteristics that enable recognition of regions/objects with singular features in regard to their neighbours. First, an input image  $\mathcal{I}$  is subsampled into a Gaussian pyramid and each level  $\sigma$  is decomposed into channels for contrast-based image features such as colour, intensity, and orientation, i.e., red (R), green (G), blue (B), yellow (Y), intensity (I), and local orientations ( $O_\theta$ ). From these channels, centre surround “feature maps”  $\mathcal{F}_l$  for different features  $l$  are constructed, as differences between “centre” fine scales  $c$  and “surround” coarser scales  $s$ , and normalized. The feature maps are summed over the centre-surround combinations using across-scale addition and the sums are normalized again.

$$\mathcal{F}_l = \mathcal{N} \left( \sum_{c=2}^4 \sum_{s=c+3}^{c+4} \mathcal{F}_{l,c,s} \right), \forall l \in L_I \cup L_C \cup L_O \quad (2.4)$$

where  $L_I = \{I\}$ ,  $L_C = \{RG, BY\}$ ,  $L_O = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ . For the general features colour and orientation, the contributions of the dimension features are linearly summed and normalized once more to yield “conspicuity” maps. Thus,

$$C_I = \mathcal{F}_I, C_C = \mathcal{N} \left( \sum_{l \in L_C} \mathcal{F}_l \right), C_O = \mathcal{N} \left( \sum_{l \in L_O} \mathcal{F}_l \right) \quad (2.5)$$

Finally, all conspicuity maps  $C_k$  are linear combined once more to generate the saliency map  $S$ ,

$$S = \frac{1}{3} \sum_{k \in \{I, C, O\}} C_k \quad (2.6)$$

Despite its simple architecture and low computational cost, the authors claim that this model performs well with complex natural scenes, and quickly detects salient traffic signs of varied shapes, colours and textures.

**Hou's model**[29] - this visual saliency model operates in the frequency domain based on the idea that similarity implies redundancy. The authors exploit statistical singularities in the spectrum as these may be responsible for anomalous regions in the image, where proto-objects come out. According to coherence theory of Rensink [31], proto-objects are candidate objects (units of visual information) which have been detected but not yet identified as an object. For an input image  $\mathcal{I}(x)$ , amplitude  $\mathcal{A}(f)$  and phase spectrum  $\mathcal{P}(f)$  are obtained from the Fourier Transform  $\mathfrak{F}$  of  $\mathcal{I}(x)$ . Then, the log spectrum  $\mathcal{L}(f)$  is processed from the down-sampled image. Then, the spectral residual  $\mathcal{R}(f)$  is obtained as  $\mathcal{R}(f) = \mathcal{L}(f) - \mathcal{A}(f)$ . The  $\mathcal{A}(f)$  can be approximated by convolving an  $n \times n$  local average filter  $h_n(f)$  with  $\mathcal{L}(f)$ . Using the inverse Fourier Transform  $\mathfrak{F}^{-1}$ , the saliency map in the spatial domain are obtained. Finally, the saliency map  $\mathcal{S}(x)$  is smoothed with Gaussian filter  $g(x)$  ( $\sigma = 8$ ) for better visual effect. The entire process to obtain the  $\mathcal{S}(x)$  can be summarised as follows:

$$\begin{aligned} \mathcal{A}(f) &= \Re(\mathfrak{F}[\mathcal{I}(x)]), \\ \mathcal{P}(f) &= \Im(\mathfrak{F}[\mathcal{I}(x)]), \\ \mathcal{L}(f) &= \log(\mathcal{A}(f)), \\ \mathcal{R}(f) &= \mathcal{L}(f) - h_n(f) * \mathcal{L}(f), \\ \mathcal{S}(x) &= g(x) * \mathfrak{F}^{-1}[\exp(\mathcal{R}(f) + \mathcal{P}(f))]^2. \end{aligned} \quad (2.7)$$

The authors used a threshold segmentation to find proto-objects in the saliency map  $\mathcal{S}(x)$ .

**Bruce’s model [30]** - Bruce *et al.* proposed the Attention based on Information Maximisation model (AIM) which utilizes Shannon’s self-information measure for computing saliency maps. In this model a saliency map of an image region is the information that such region conveys relatively to its neighbour regions. Information of a visual feature (VF) is defined as  $I(VF) = -\log p(VF)$  where,  $p(\cdot)$  is the probability density function.  $I(VF)$  is inversely proportional to the likelihood of observing VF (i.e.,  $p(VF)$ ). To calculate  $I(VF)$ , the  $p(VF)$  must be firstly estimated. In the computation of the  $p(VF)$  a histogram over small image regions is used. In RGB images, considering a local patch (image region) of size  $M \times N$ ,  $VF$  has the high dimensionality of  $3 \times M \times N$ . Since a single image has insufficient data to produce a reasonable estimate of probability distribution  $p(VF)$ , a representation based on Independent Components Analysis (ICA) is used. Finally, at each image location, the probability of observing the RGB values in a local image patch is the product of the corresponding ICA basis likelihoods for that patch.

### Saliency detection using temporal information

The main aim of the visual saliency computation methods based on temporal information is to separate motion regions from the background, given that viewers tend to direct their gaze towards higher motion regions [32]. Several motion-based methods are extensions from spatial domain. For example, Temporal Spectral Residual (TSR) [33] is an extension of the Hou’s model [29]. The TSR approach is based on the underlying idea that background motion is usually smaller and more regular than foreground motion, the foreground will form a distinct trajectory from the background. Thus, TSR approach estimates salient regions by removing superfluous information. TSR algorithm used the Hou’s model on video slices along X-T and Y-T planes to distinguish foreground motion regions from backgrounds. Then, a threshold selection scheme is used to reject noise. Finally and to refine the results, a voting scheme is applied to obtain final saliency map.

Qureshi [34] proposed the Temporal Image Signature (TIS) which can be seen as an extension of the Image Signature (IS) method [35] to the temporal domain. TIS combines IS and TSR methods [33] for computing the visual saliency map. The Discrete Cosine Transform (DCT) is used to estimate the salient regions. Qureshi’s methods can be divided into three main steps, frame division, saliency detection, transformation and accumulation. In the frame division, a video is sliced into X-T and Y-T planes, this strategy was also used by [33] in the TSR method. Saliency detection uses the DCT information to identify the saliency movement. Finally, the saliency map is obtained by accumulating the transformation back from the X-T and Y-T to XY domain.

### Saliency detection using depth information

In the literature, the main contributions to saliency detection are focused on 2D image/video processing. To extend saliency computation methods to 3D images and video it is necessary to include depth-related information in the model. Most of the existing methods to compute saliency maps in 3D images combine visual features extracted from spatial information (luminance, colour, orientation or intensity) with some depth-related information. Maki *et al.* [36] proposed a visual attention model which includes image flow, the motion direction and stereo disparity for region identification. The authors used the depth information to prioritize image regions, assigning higher priority to those objects that are closer to the viewer. However, in the context of video content analysis, the near targets/objects are not necessarily the most important or salient. Only qualitative assessment, based on author's description, was used to validate the method.

Ouerhani and Hugli proposed a visual attention model using depth and 2D visual features to produce the saliency map [37]. First, the feature maps are built with multiple features extracted from the texture and depth information, such as intensity, colour components, intensity gradient components, raw depth and other depth related features. Each feature map is transformed into a conspicuity map based on a multi-resolution centre-surround mechanism and then a linear combination of conspicuity maps, produces the saliency map for each 3D image. Although several 2D and depth features were considered in the description of the method, the experimental validation is insufficient because it only uses depth and colour. Other limiting aspect of the work is that no comparison to eye-tracking data is done to validate the results.

Zhang *et al.* [38] proposed a bottom-up visual attention model for stereoscopic content where the depth map is combined with motion and a static saliency map computed using Itti's model [22]. The authors considered that pixels closer to the viewers and in front of scene are more salient. The final saliency map is obtained from these three attributes with arbitrary weights. However, the authors did not validate their method and there was no comparison between the saliency maps and other methods or fixation density maps. Another type of research was carried out by Patapova *et al.* [39] using probabilistic models of various 2D (colour, orientation and intensity) and 3D cues (surface height, relative surface orientation, occluded edges). These are then fused with a linear combination scheme to obtain the final saliency map. This method was developed for robotic applications.



In a recent work, Wang *et al.* [2] proposed a visual attention model for stereoscopic images, which combine the 2D and depth saliency maps. The main difference between this model and others mentioned before, is the use of a processing step for extracting features into a depth saliency map, which is followed by a merging operation that combines the depth saliency map with the 2D saliency map. The final visual saliency map for 3D images is equal to sum of both saliency maps. Wang *et al.* used the Itti’s model [22], the AIM model from Bruce [30] and Hou’s model [29] to predict 2D saliency maps and proposed a new depth saliency map generation based on a probability-learning from eye-tracking data. In this work, the author used the depth contrast as a feature to produce the depth saliency map, by applying a Difference of Gaussians (DOG) filter to the depth map to extract the depth contrast. The author also created a database, for the model validation, which is publicly available, containing stereoscopic images, disparity maps and eye-tracking data.

More recently, Jiang *et al.* [40] proposed a visual attention model for stereoscopic image quality assessment tasks, based on a 2D saliency model, centre-bias, depth cue (foreground cue and background). The final stereoscopic saliency map is obtained by adding individual perceptual features. In their work, the authors use a saliency map as a modulation function to derive an image quality assessment metric. Iatsun *et al.*, [41] presented a saliency model for 3D video based on the fusion three saliency features of the different dimension, spatial, temporal and depth. Contrary to most 3D saliency models proposed in the literature, the authors proposed a comprehensive qualitative evaluation in this work, using eye-tracking experiments to validate the proposed model.

### 2.2.2 Performance metrics

Performance metrics to evaluate the “goodness”, or quality, of visual saliency maps created by computational models are still an open research issue for 3D video. However, in the case of 2D content there are several methods widely used to evaluate the quality of saliency maps computed from different models. Among the most relevant, one can find the use of Pearson Linear Correlation Coefficient (PLCC) [42, 43], Kullback-Leibler Divergence (KLD) [42], Area Under the receiver operating characteristic (ROC) Curve (AUC) [30, 43, 44] and Normalized Scanpath Saliency (NSS) [45]. While the first three are directly applicable to saliency maps and fixation density maps, NSS compares the fixation map with a saliency map.

In the following, the four objective metrics with more consensus from the literature are

explained with more details. However, some research works use visual comparison of their saliency maps with saliency maps produced by other methods and fixation density maps obtained from eye-tracking experiments [30].

### Pearson linear correlation coefficient

In this context, PLCC provides a similarity measure of two saliency maps using a single scalar value. The PLCC ranges between  $-1$  and  $+1$  and values close to  $0$  indicate poor correlation between the two maps. Values close to  $1$  or  $-1$  indicate a high correlation and the sign gives an indication of the phase of the variables variations, i.e., whether they vary in the same or in opposing directions. The PLCC is implemented as a measure of the linear correlation between the computed saliency map  $S$  and the corresponding fixation density map  $F$ , as follows,

$$PLCC(S, F) = \frac{cov(S, F)}{\sigma_S \sigma_F} \quad (2.8)$$

where  $cov(S, F)$  is the covariance and  $\sigma_S$  and  $\sigma_F$  are the standard deviation of the  $S$  and  $F$  respectively.

### Kullback-Leibler divergence

The Kullback-Leibler divergence is used to compute the degree of dissimilarity between two Probability Density Functions (PDF). In this context, the computed saliency map and the corresponding fixation density map ( $H$  and  $P$ ) respectively, are necessary for comparison. The  $H$  and  $P$  are transformed into two probability density functions  $h_x$  and  $p_x$ , which are estimated using kernel density estimation (e.g., *ksdensity* function in Matlab). The Kullback-Leibler divergence, noted KLD, between  $H$  and  $P$  is given by the relative entropy of  $H$  with respect to  $P$ :

$$KLD(H, P) = \sum_x h_x \ln \left( \frac{h_x}{p_x} \right) \quad (2.9)$$

The KLD is defined only if  $h_x$  and  $p_x$  both sum to  $1$  and if  $h_x > 0$  for any  $x$ , such that  $p_x > 0$ . The KLD is not distance, once it is not symmetric and does not satisfy the triangle inequality, i.e.,  $KLD(H, P) \neq KLD(P, H)$ . The KLD is nonlinear and varies in the range of  $[0 \dots \infty]$ . When the two probability density functions  $h_x$  and  $p_x$  are strictly equal, the KLD value is  $0$ .

### Normalized scanpath saliency

NSS was proposed by Peters *et al.* in [45] to measure the correspondence between eye fixation location and the computed visual saliency maps, taking into account the variability of eye movements. In this method, each saliency map  $S$  is first normalized to have zero mean and unit standard deviation, according to:

$$N_S = \frac{S - \mu_S}{\sigma_S} \quad (2.10)$$

where  $N_S$  is the normalized saliency map,  $\mu_S$  and  $\sigma_S$  are the mean and the standard deviation of the computed saliency maps  $S$  respectively. Then for each point corresponding to the fixation locations, the normalized salience value is extracted and the mean of all these extracted values is calculated. This mean is the normalized scanpath salience value, i.e., NSS.

The NSS values can be: (a)  $NSS = 0$ , there is no correspondence between human fixation (eye positions) and computed salient regions; (b)  $NSS < 0$ , this indicates an anti-correspondence between human fixation locations and computed salient points, i.e., the eye positions are on non-salient regions; (c)  $NSS > 0$ , when the eye positions are projected on the salient regions.

### Area under the curve

Using this metric, the computed saliency map  $S$  has to be “thresholded” into a binary mask  $S_{bin}$  as,

$$S_{bin}(i, j) = \begin{cases} 1 & \text{for } S(i, j) \geq \tau \\ 0 & \text{for } S(i, j) < \tau \end{cases} \quad (2.11)$$

where  $S_{bin}(i, j)$  is the binary mask of computed saliency map  $S$  for each pixel  $(i, j)$ . Using this representation, the pixels with larger saliency values than a threshold  $\tau$  are classified as fixated, while the rest of the pixels in that image are classified as non-fixated.

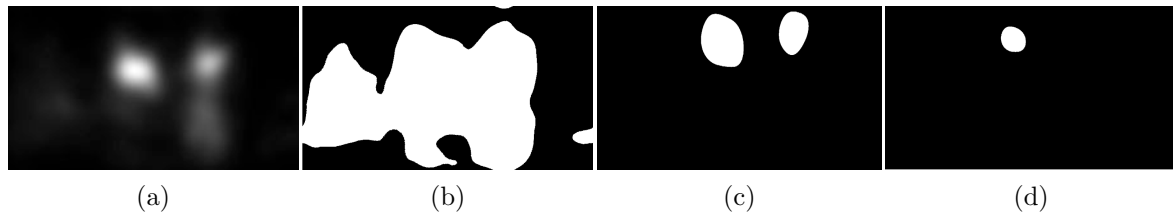


Figure 2.9: Computed saliency map of the *News report* sequence (a)  $\tau = 10$  (b)  $\tau = 100$  (c)  $\tau = 200$  (d).

### 2.2.3 Applications

There are many applications of visual attention models which have been developed for quite diverse fields, such as robotics [46], advertising [47], medicine (finding tumors in mammograms [48], retinal prosthesis [49]) and computer vision. There are relevant areas of application in image and video segmentation [50], image and video quality assessment [51], image and video coding [52, 53], salient object detection [22, 29], object recognition [54], ROI detection [55–57], scene classification [58], video summarisation [59–64], video shot detection [65] and image and video retargeting [66]. In this section, more details to applications are given somehow related to the work of this Thesis.

## 2.3 Video retargeting

This section presents an overview of the video retargeting methods, as these are also used to produce flexible representation of video content to match lower resolution requirements. Thus, in general such methods aim to resize an original video sequence to desired resolution or aspect ratio. Different methods are described including non-content-aware and content-aware retargeting methods. The performance metrics used to evaluate retargeting methods and some retargeting's applications are also addressed.

Video display devices have various resolutions and aspect ratios (e.g., 16:9, 4:3 and 3:2). Thus, watching video with the improper aspect ratio decreases the quality of experience, producing visual discomfort or important information is lost. When the size or the aspect ratio of the target display is different from the original one, the visual content must be adapted to the target display. A simple way to make this adaptation is adding a black bar or linear scaling (see Figure 2.10), however these approaches would bring some unpleasant viewing quality and some times do not preserve important visual content. For example, displaying a football match on a mobile device is a good example for the need of a smart adaptation method, while the full high resolution may render the players and the ball too small, it would be better to display less content where the ball remains large enough to be easily seen. In this case a video retargeting method for adapting the visual scenes of a football match to a lower resolution display, must use structural and semantic information of the input video to obtain a better representation of the original content i.e., important objects of each frame should be kept, along with temporal smoothness and coherence as well as low visual distortion. In the next sections, different video retargeting methods are described with some emphasis on their capability of producing relevant representations of the original visual content, at a lower resolution.

### 2.3.1 Non-content-aware video retargeting

A simple approach to implement video retargeting is to modify the size and the location of a user-defined cropping window user-defined manually and frame-by-frame. However, this approach is time consuming and not adapted to live events which require short delay between acquisition and display. Cropping and linear scaling are two automatic solutions to performance change the aspect ratio of a given video signal, but despite of their simplicity and easy implementation they are content agnostic and do not consider

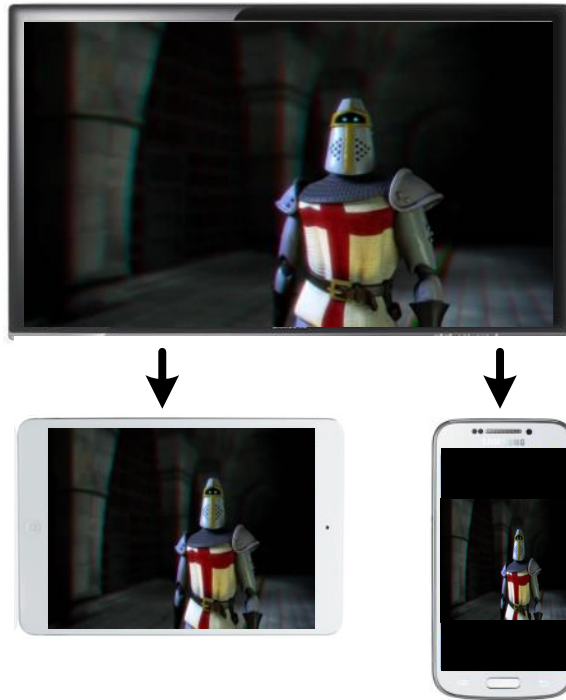


Figure 2.10: Examples of image retargeting for devices with different display resolutions.

any type of user-driven approach.

Linear scaling methods can be based on reduction (downsampling) or increase (upsampling) of number of pixel that are used to represent video content. Figure 2.11b shows the result of linear scaling applied to the *Jockey* sequence, using the MPEG-4 down-sampling filter to retarget the video content from higher to lower resolution. Non-content-aware cropping selects a window of some predefined size  $n^* \times m^*$  from the original video frame of size  $n \times m$ , where  $n^* \leq n$  and  $m^* \leq m$ . The content inside the window is kept whilst outside content is discarded. In centred cropping method the selection window is selected from the centre of video frame, assuming that the ROI is located in the centre of scene. However, whenever this principle is not valid the output fails, as it can be seen in Figure 2.11c.

### 2.3.2 Content-aware video retargeting

Content-aware video retargeting methods aim to produce a representation of the visual content taking into account image regions with different relevance in order to choose those that are more relevant for the users, according to some predefined criteria. Several

(a) Original ( $3840 \times 2160$ )(b) Linear scaling ( $1280 \times 720$ )(c) Centred cropping ( $1280 \times 720$ )

Figure 2.11: Non-content-aware cropping and scaling methods.

studies of video retargeting use image-based retargeting methods with different types of adaptation to maintain the temporal coherence [67–70]. Most of these methods are composed by two steps, the first is ROI detection and then cropping and scaling to the target resolution. The various video retargeting methods can be grouped into: cropping, seam carving, warping and hybrid methods (combination of the different retargeting methods).

### Cropping

The cropping methods are based on the principle that regions of interest must be preserved while the content in the borders of the frame can be dropped. Visual saliency maps, optical flow, or relevant objects, e.g., faces or moving objects, may be used to determine regions of interest. Fan *et al.* [71], used together a visual attention model based on motion, face, saliency and text to detect the regions of interest and a virtual camera control system

to crop the original video. Wang *et al.* [72] also presented a system for retargeting video to tiny devices based on visual attention of the frame and used cropping to remove the less important content. Kopf *et al.* [73] used the foreground objects like general objects, faces and superimposed text as regions of interest. More recently, the video cropping method presented in [74] use temporal constraints to smooth the trajectory of the cropped windows.

### Seam carving

Seam carving was firstly proposed in [75] for image retargeting, by reducing the image size through removal of minimum energy seams on vertical or horizontal connected path. In this first implementation of seam carving, the image size is changed according to its content, but saliency regions are not considered, which causes artefacts in large objects. This implementation uses a gradient-based energy, which highlights only distinctive edges. However, several works were proposed to solve this limitation of the early implementations by enhancing energy functions. Rubinstein *et al.* [68] extended image seam carving to video retargeting by imposing temporal constraints. Kolf [69] and Grundmann [70] adapted existing image seam carving methods to keep temporal coherence in video. Yan *et al.*, [76] present a motion aware seam carving solution to preserve the temporal smoothness of seams. More recently, Wang *et al.*, [77] proposed a shape matching solution to protect the shapes of the salient curves from deformation. Although seam carving methods may produce visually pleasant results in some particular cases, their generic application in non-constrained scenarios seem to far from producing acceptable results in all types of visual content.

### Warping

Warping based video retargeting methods extend image warping methods to video sequences. In the warping-based methods, an image is first segmented into regions (e.g., grids), which are then non-uniformly deformed through an optimisation process, in which spatial constraints are imposed to minimize the deformation distortion on important pixels/regions. Wolf *et al.*, [67] used non-uniform global mesh warping scheme for video retargeting. Recently, Li *et al.*, [78] presented a video retargeting method in which the consistency of the content is kept, for that the method divides a video into spatio-temporal grids, called grid flows. Theses grids are used to select the key-frames and then resized these frames via quadratic programming.



## Hybrid methods

In addition to the methods previously presented, some researchers investigated some combinations of different methods. For instance, Liu *et al.*, [79] used video content to determine the best combination of cropping and scaling methods to match the video frame resolution to the target display. Warping-based methods were combined with cropping by the Wang *et al.*, [80, 81], also for video retargeting. More recently, Kiess *et al.*, [82] combine seam carving and cropping methods for real-time adaptation of video.

### 2.3.3 Performance metrics

Performance metrics to evaluate video retargeting methods should measure how the most important content is preserved and also whether the original video content is maintained. Furthermore, these metrics must quantify how retargeted video is free from visual artefacts. Until recently, there is not a clear definition or well-known metrics to objectively evaluate the quality of retargeted video. In an attempt to respond to these limitations several metrics were proposed and they can be classified into three types: result description, subjective and objective. Usually, more than one metric is used to evaluate retargeting methods [76–78, 80, 81].

**Result description** - it is the most common quality metric used to evaluate retargeting methods. This is often based on a descriptive approach, employed to clarify and describe the benefits of some method in comparison with others. This evaluation metric is used in [68, 70, 74, 76–81] here a visual comparison of the retargeted video is realised. Wang *et al.* [77] used the visual comparison of images obtained with frame difference to evaluate temporal consistency of their video retargeting method. The frame difference of the original frames (temporally coherent) are compared with the frames difference obtained by retargeting methods under analysis.

**Subjective methods** - it is widely adopted for video retargeting methods [69, 71, 72, 76, 78, 80]. Fan and Jun Wang [71, 72] used subjective assessment questions to evaluate their retargeted method. Kolf [69] applied two approaches to evaluate his retargeting method. In the first, the viewers evaluated his proposed method by watching the original video first and the other retargeted versions. Then, a questionnaire with open/close questions is filled out, (“How well are details preserved?”, “What kind of disturbing effects did

you recognize?”, “Which visual errors did you recognize?” and “What is your overall impression of the adapted video?”). The second task is to sort the retargeted video by the visual quality. More recently, several authors proposed different approaches based on double and single stimulus evaluation methods, specified by ITU-R BT.500-11 [83], for assessment of the proposed methods. The retargeting solutions published in [76–78, 80] used a subjective evaluation method in which the original video and the pairs of retargeted video are shown at the same time to viewers and then each subject, is asked to choose one of the two retargeted versions according to his/her preference.

**Objective methods** - objective assessment through computational methods was used by several authors [69, 78, 80, 81] to compare their proposed methods with others. For instance based on the eye tracking data, Chamaret *et al.* [84] proposed an objective metric to assess the quality of a video retargeting methods. Although fast and efficient computation is easy to achieve the quality metrics and models for objective evaluation of retargeting methods are kept under investigation to better match the quality of experience.

### 2.3.4 Applications

There are several application scenarios where video retargeting may be useful. Nowadays, High Definition TV (HDTV) is widely used, so video retargeting may be required to improve content mapping from 4:3 aspect ratio to 16:9 wide screens. Mobile communications and pervasive wireless internet access also need content adaptation requirements due to the display size of mobile devices, which is typically smaller than a TV or computer monitor. Thus, retargeting methods are required to adapt the visual content to the display size of such mobile devices. In urban public facilities (e.g., entertainment zones, LED stadium walls, and digital bulletin board), video retargeting is also needed for content adaptation to different displays.

#### Video summarisation

Computational methods for video summarisation through key-frame extraction based on attention models were first proposed by Ma *et al.* in [59] for 2D video, using visual, aural and linguist attention features combined by a non-linear fusion scheme. The key-frames selection process is based on the location of the peaks of an attention curve computed along the video sequence. Peng and Xiao-Lin [85] also proposed a key-frame extraction method

based on a visual attention model designed to compute dynamic and static salient features from 2D video. A drawback of these works is that no systematic methodology was used to evaluate the results, either subjectively or objectively. A different approach was followed by Lai and Yi [62], using a saliency-based visual attention model to extract the key-frames from a 2D video sequence. Dynamic and static conspicuity maps were built based on motion, colour and texture features and then these conspicuity maps were combined to generate an attention curve from which key-frames are identified. Another method based on similar principles was presented in [63], where both static and dynamic visual attention values are non-linearly fused into an attention curve for key-frame extraction. More recently, the same author proposed a visual attention scheme that merges low level features and motion intensity for extracting key-frames from 2D video [64]. However the visual attention model used by the authors is not validated with ground-truth data, i.e., no comparison with eye-tracking data or similar is done.

### **Image and video retargeting**

Image and video retargeting methods have been gaining interest from the research community due to the massification of mobile devices with the capacity to play video and games using a great variety of screen sizes. Several methods can be used to adapt high resolution video to small screen devices, such as linear scaling (i.e., down-sampling), image cropping, seam carving and warping. Depending on the target application and device, each approach has its own advantages and limitations. For instance, after downsampling, important details in the original scene may be no longer recognized. Retargeting methods based on content-aware were proposed for example cropping, seam carving, warping and hybrid methods (combination of the different retargeting methods). More details of these retargeting methods can be found in the Section 4.1.

### **Perceptual video coding**

Combining the non-uniform visual attention of human observers with the limited region around the centre of eye fixations that can be seen at high resolution, while the rest of the image is blurred, opens the possibility of non-uniform coding based on perceptual features to increase coding efficiency. For instance, Itti [52], proposed a compression method based on attention, where regions of interest are chosen based on a non-linear integration of low-level visual cues. A dynamic foveation filter then blurs every frame,

increasingly with distance from salient locations and finally the resulting sequence is coded using H.264. More recently, Guo and Zhang [53] proposed spatio-temporal saliency computation method called Phase Spectrum of Quaternion Fourier Transform (PQFT). The authors showed that, the proposed method can improve coding efficiency in image and video compression.

## 2.4 Video summarisation

Video summarisation is a content description technique which provides high level of flexibility in the representation of the video content in the sense that only the most important segments of a video sequence can be selected from a very high degree of conciseness (e.g., single key-frame) to short subsequences with variable length. The basic concepts of video summarisation and its intrinsic processes of shot-boundary detection, key-frame extraction and key-frames presentations methods are reviewed in this section. Performance metrics and some multimedia applications are also pointed out.

Video summarisation automatically creates a short version of an original sequence, i.e., a video segments or subset of salient frames, chosen as essential to represent the original video content according to some predefined criteria. This selection can be based on the analysis of video content's features like colour, motion, and audio [86], or specific information previously created like MPEG-7 description [87]. Thus, the user can get an idea of the relevant content in the video sequence without having to see the whole sequence. However, video summarisation introduces distortions at the playback stage and this distortion is related to the conciseness of the summary whereby a more succinct summary implies higher distortion. Video summary generation methods can also be implemented on either uncompressed or compressed video [88].

The literature defines two types of video summaries, namely those based on key-frames and those comprised of video skims [86]. A video summary based on key-frames is made up of a set of relevant frames selected from the video shots obtained from the original video. This type of summary is static, since the key-frames, being temporally distant and non-uniformly distributed, do not enable the rendering/reproduction of the original temporal evolution of the video. Here, the video content is displayed in a quick and compact way, without timing or synchronisation requirements for browsing and navigation purposes. Video skims are usually built by extracting the most relevant temporal segments (with or without audio) from the source sequence. After the extraction, all temporal segments

are concatenated into a video with much shorter length than the source sequence. The computation of the key-frames and video skins summaries are distinct, but these two video-content representations can be transformed from one to the other. The video skins can be generated from key-frame summary by adding frames or segments that include the key-frame and a key-frame summary can be created from a video skim by uniform sampling or by selecting one frame of each video skim segment [89].

In the available literature one can find several recent methods for summarisation of 3D video. However, a comparative study between different methods was not done so far. In the scope of this Thesis a review of video summarisation methods based on key-frames was carried out to better understand the related issues and also highlight the most important characteristics of each one. The key-frame extractions methods, key-frame presentation, evaluation and application used in 2D video are also utilized and extended with some adaptation in 3D video. This overview of the current state-of-the-art is mainly focused on the methods and features that are used to generate and evaluate key-frame summaries and not so much on the limitations or performance of specific methods. Since experimental set-ups, features used for summarisation, and 3D video formats are considerably different from method to method, a detailed comparative analysis of the results, advantages and shortcomings of all methods is not possible with the available data.

Most of the key-frame summarisation methods presented in the literature are based on a three-step approach: first, the entire video sequence is divided into video shots based on scene transitions using a Shot Boundary Detection (SBD) method that matches the application requirements. Then, a key-frame extraction method is applied to each video shot to extract the most representative frames, based on specific properties of the video content and a similarity measure. Finally, the extracted key-frames are presented to the viewers or stored in a container following some predefined presentation structure.

A conceptual framework for key-frame summarisation is shown in Figure 2.12. SBD, key-frame extraction and key-frame presentation are the three main stages of the framework. The input video is segmented into video shots, mostly based on spatio-temporal criteria, but other criteria can be used such as based on motion [8, 90] or the combination of the temporal and depth features [91]. More details can be found in the next section. After segmentation, one or more key-frames are extracted from each video shot according to user-defined parameters or based on specific requirements. The most relevant key-frame extraction methods are presented in Section 2.4.2. Once the key-frames are extracted, they need to be presented in an organized manner for easy viewing during video browsing or

navigation. In this framework, three key-frame presentation methods are described, static storyboard, dynamic slideshow and single image based on stroboscopic effect, but other methods can be found in the literature (see Section 2.4.3). The key-frame presentation methods are independent of the key-frame extraction operation and thus the same key-frame summary can be presented to viewers in different ways.

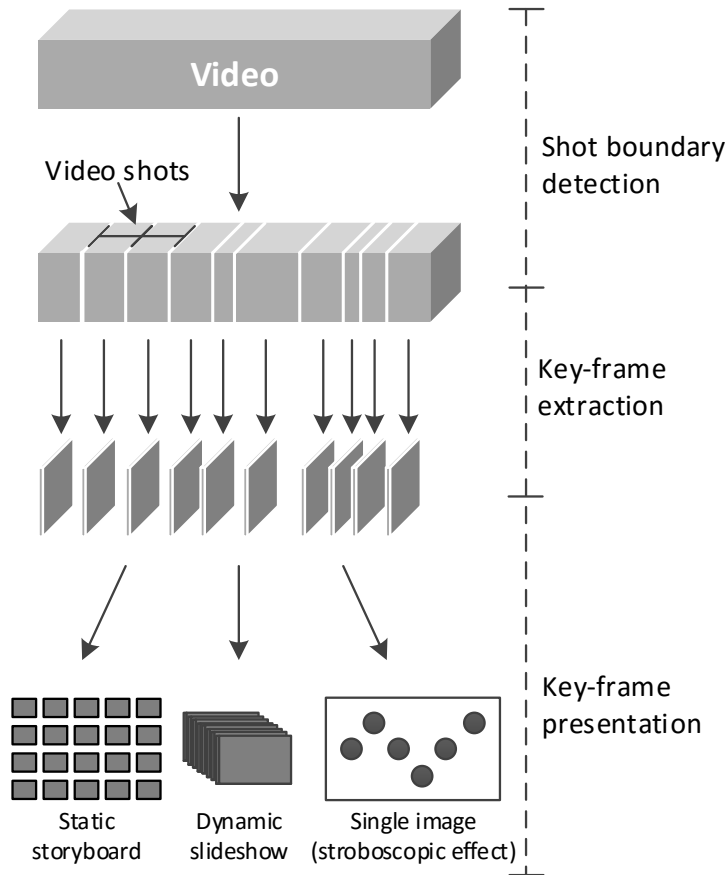


Figure 2.12: A conceptual framework for key-frame summarisation.

### 2.4.1 Shot boundary detection

In the recent past, development of SBD methods for 2D video received a lot of the attention from the research community. However, very few works have investigated the SBD problem in the context of 3D video, especially taking into account depth information. Relevant surveys of video SBD methods with specific application in 2D video can be found in the literature [92–94]. In this section, it is introduced the main concepts behind these

methods for 2D video. Then, the most promising and better performing SBD methods used for 3D key-frame extraction are explained in detail.

A video segment can be decomposed into a hierarchical structure of scenes, video shots and frames, with the linear video first divided into video scenes, which may comprise one or more video shots (set of correlated frames). A video shot is defined as a set of frames which is continuous and temporally and spatially cohesive [95]. Thus, the video shot is the fundamental unit in the content structure of a video sequence. Since its size is variable, the identification of start and end of the video shots is done using specific SBD methods.

Figure 2.13 presents a generic framework of SBD methods, comprising three main steps. Firstly, visual features are computed for each video frame. In this step, the pixel intensities, colour histograms, image edges, camera operations, or coded information such as DCT coefficients, DC terms, motion vectors, MB coding modes and bit rate can be used [92]. In the second step the visual features of consecutive frames are compared and some decision method is used to identify shot boundaries. The decision methods used to find shot boundaries can be based on static thresholds (as in Figure 2.13), adaptive thresholds (thresholds depend on the statistics of the visual features used), B-splines fittings [96], Support Vector Machines (SVM) [97] and K-means clustering [91]. The detection accuracy of SBD methods is improved by combining several visual features [98].

Video shot boundaries can be classified into two types: Abrupt Shot Boundary (ASB) and Gradual Shot Boundary (GSB). In ASB the scene transition occurs over very few frames, usually a single frame defines the boundary. In the case of GSB, the transition takes place gradually over a short span of frames. The most common gradual transitions are fade-ins, fade-outs, dissolves and wipes [92–94]. A common problem in SBD is the correct discrimination between camera operations and object motion that originate the gradual transitions, since the temporal variation of the frame content can be of the same order of magnitude and take place over the same number of frames. This similarity of visual effects caused by camera operations and object motion can induce false detections of gradual shot boundaries. This problem is aggravated for video sequences with intense motion.

### **SBD methods for 3D video**

Doulamis *et al.* in [99] proposed a key-frame extraction method for stereo video which includes a SBD method. Here, the entire video sequence is divided into video shots using

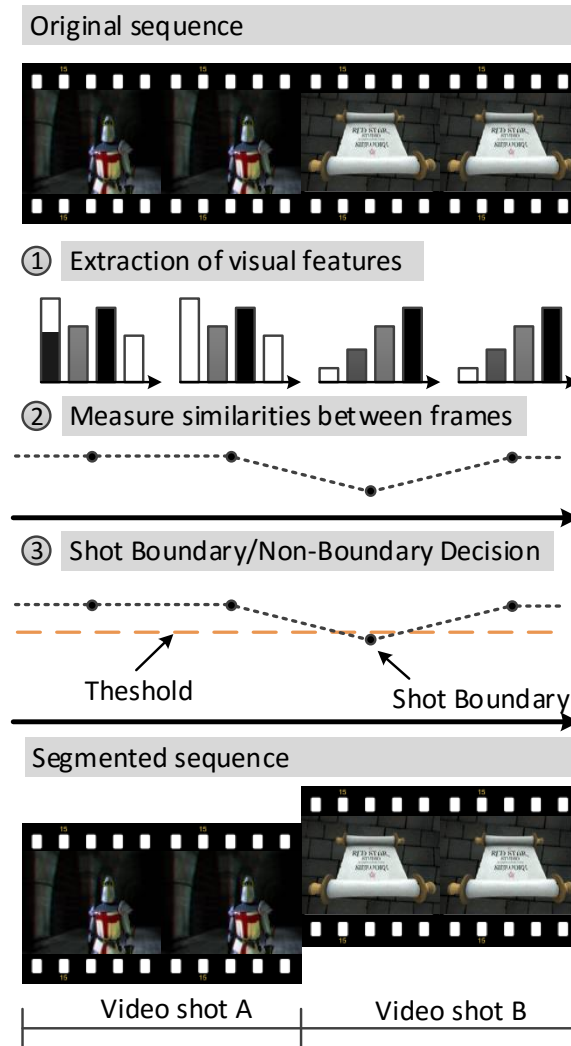


Figure 2.13: A generic diagram of SBD framework.

an algorithm based on the analysis of DC coefficients of compressed videos, following the solution proposed in [100]. More recently, Papachristou *et al.* in [101] presented a framework for stereoscopic video shot classification, that uses a well-known method designed for 2D video to segment the original stereoscopic video into shots [102]. However, this method was applied only to the colour channels of the videos to be summarised. Ferreira *et al.* [91] proposed an algorithm to detect 3D Shot Boundaries (3DSB) based on a joint depth-temporal criterion. The absolute frame difference and sum of absolute luminance histogram difference are used as the relevant measures in the temporal dimension, while in the depth dimension, the variance of depth in each frame is used. A K-means clustering algorithm that does not require training and does not use thresholds is applied to choose



the 3DSB transition frames. Ferreira's method is independent of the video content and can be applied to 2D or 3D video shot boundary identification. In the case of the 2D video, absolute frame differences and sum of absolute luminance histogram difference are used.

Some methods target segmentation of 3D mesh sequences using properties of 3D objects as the shape and motion/action (e.g., human body motion, raise hands) to detect the shot boundaries. Yamasaki *et al.* [90] proposed a temporal segmentation method for 3D video recordings of dances, which is based on motion speed, i.e., when a dancer/person changes motion type or direction, the motion becomes small during some short period and in some cases it is even paused for some instants, according to the type of dance. To seek the points where motion speed becomes small, the authors used an iterative close point algorithm proposed in [103] which is employed in the 3D space (spherical coordinates). In contrast to conventional approaches based on thresholds, the authors devised a video segmentation scheme appropriate for different types of dance. Since the decision rule is not based on absolute values and thresholds, rather on relative values of extrema, it is more robust to data variation (like type of dance) and no empirically derived decision thresholds are used.

Another method which uses the motion speed of the 3D objects was presented by Xu *et al.* [5]. To reduce computation time of motion information, the authors used the point distance (DP) instead of vertices position in Cartesian coordinates. DP is defined as Euclidean distance between one fixed point and all 3D objects' vertices coordinates of each frame. Figure 2.14 shows the point distance for two frames of Batter's sequence. Before determination of scene transitions, the histogram of point distance of each frame is calculated.

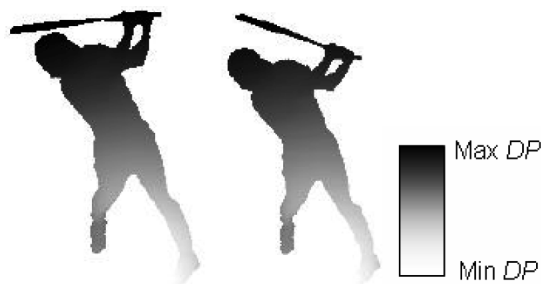


Figure 2.14: Point distance of the frame #38 and #39 of the *Batter* sequence. Grey values means the point distance from (0,0,0) [5].

To detect abrupt and gradual transitions of 3D video the Euclidean distance between the

histograms of point distance and three thresholds are used, where the threshold values were derived empirically.

Ionescu *et al.* [104], used a histogram-based algorithm specially tuned for animated films to detect ASB. From GSB only fades and dissolves are detected, since they are the most common gradual transition. The GSB detection is done using a pixel-level statistical approach proposed by [105]. The authors proposed the Short Colour Change (SCC) detection algorithm to reduce the false positives of cut detection. The SCC is the effect that accompanies short term frame colour changes, caused by explosion, lightning and flash-like visual effects. More recently, Slama *et al.* [106] proposed a method based on the motion speed to split a 3D video sequence into segments, characterised by homogeneous human body movements (e.g., walk, run, and sprint). However, the only indicators that the author considers as significant video shot transition are changes in type of movement. Moreover, video shots with small differences from previous shots and small number of frames are avoided. The motion segmentation used in this work is based on finding the local minimum of motion speed to detect the breakpoint where the human body movements change and use these changes to segment the entire video into shots.

### Performance metrics

Three well-known performance indicators are used in the evaluation of the SBD methods for 2D video: Recall rate (R), Precision rate (P) [107] and accuracy measure F1 [108]. The computation of these values is based on the comparison of manual segmentation (ground-truth) and computed segmentation. If a ground-truth is available then these metrics can be applied to 3D video SBD methods.

Recall rate is defined as the ratio between the number of shot boundaries detected by an algorithm  $D$  (i.e., correctly detected) and the total number of boundaries in the ground-truth dataset (sum of  $D$  and the number missed boundaries  $D_M$ ) as given by Equation (2.12). Precision rate, computed according to Equation (2.13), is defined as the ratio between the number of shot boundaries detected by an algorithm and the sum of this value with the number of false positives  $D_F$ . F1 is a measure that combines P and R, see Equation (2.14).

$$R = \frac{D}{D + D_M} \quad (2.12)$$

$$P = \frac{D}{D + D_F} \quad (2.13)$$

$$F1 = \frac{2RP}{R + P} \quad (2.14)$$

For good performance, the Recall and Precision rates should have values close to 1. The best performance is reached when F1 is equal to 1, whilst the worst occurs at 0. The Recall rate, Precision rate and measure F1 were used to evaluate the performance of temporal segmentation methods for 3D video in [5, 91, 106], while Yamasaki *et al.* [90] only used Recall and Precision rates in the evaluation process. Although, these 3D SBD methods used the same evaluation metrics, the comparison of the results and performance obtained from such SBD methods is not possible because different datasets were used.

## Discussion

Since the major difference between 2D and 3D video is the implicit or explicit availability of depth information, the visual features used in the SBD methods for 3D video must take depth into account, i.e., the temporal segmentation must also consider depth information in order to use depth discontinuities in shot detection. Until now, most research works on SBD for 3D video, have not used the depth information in the detection process. For example, Doulamis *et al.* in [99] proposed a key-frame extraction method for stereo video which includes a SBD method, but the algorithm does not take into account the depth information of the stereo video and it is only applied to one view of the stereo sequence, for instance the left view. Another drawback of Doulamis' work is the lack of performance evaluation of the proposed temporal segmentation method. Another method to segment stereo video was proposed in [101], but the proposed procedure does not take depth into account either.

In [5, 90, 106, 109], the authors proposed SDB methods for 3D video, which are only applicable to 3D mesh models and require modifications to be used with most common pixel-based 3D video formats, like stereo or video-plus-depth. Finally, Ferreira *et al.* [91] proposed a method which uses the depth and temporal information for automatic detection of 3D video shots from the 3D video sequence based on the K-means clustering algorithm to locate the boundaries. This algorithm has the advantage of not using any explicit thresholds or training procedure.

A common problem with the 2D video SBD methods described in the literature is the lack of common comparison grounds, as few works use the same dataset to test the news methods and evaluate their performance. This is a serious problem as it limits the

significance of comparisons that can be made between different SBD methods. For the 3D case, the lack of comparative analyses is even more severe, due to the reduced number of SBD methods developed so far for this type of visual information. The few works that have been proposed for SBD in 3D video usually use the Recall and Precision rate to evaluate performance, but the lack of benchmark 3D video sequences with ground-truth shot segmentations severely limit the number and types of performance evaluations that can be made. As mentioned above, the evaluation metrics are based on comparison between manual and computed segmentation. Therefore, besides being very important to have common test datasets, the development of universal and objective measures, which are specific for SBD and generic enough to be applied in different context domains and 3D video formats is highly recommended and desired.

## 2.4.2 Key-frame extraction methods

This section addresses the main concepts behind existing key-frame extraction methods and describe some relevant methods for 3D video. The key-frame extraction methods under review are grouped into seven categories: Non-optimized, Clustering, Minimum Correlation, Minimum Reconstruction Error (MRE), Curve Simplification, Matrix Factorisation and other methods.

### Non-optimized methods

The simplest method for key-frame summarisation is Uniform Sampling (UnS). This method selects key-frames at regular time-intervals (see Figure 2.15 a)), e.g., selecting one video frame every minute to be a key-frame. This results in a set of key-frames evenly distributed throughout the video sequence. However, the selected key-frames might not contain meaningful or pertinent visual content or there may be two or more similar key-frames. For instance, the selected key-frame might show a bad image (e.g., unfocused) or no key-frame exists for some video shots, thus a meaningful representation of the video content is not guaranteed.

Another simple and computationally efficient frame selection method is Position Sampling (PoS). In PoS, once the boundaries of a video shot are detected, the method selects frames according to their position in the video shot, and e.g., the first, or the last or the middle frame of the video shot (see Figure 2.15 b)) can be chosen as key-frames. Thus, the size

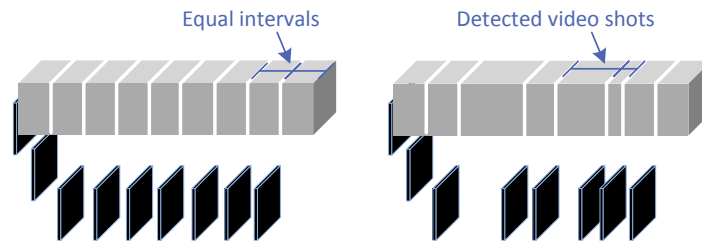


Figure 2.15: a) UnS-method: uniform sampling at equal intervals. b) PoS-method: selecting the first frame of each video shot.

of key-frame summary corresponds to the number of video shots of the entire video. In some summarisation applications one key-frame per video shot is not enough, and the PoS method can be adapted by allowing selection of multiple frames at fixed positions within the video shot. For 3D video, UnS and PoS are used mostly as references for comparisons with other methods, as in [110–112]. Ionescu *et al.* [104] selected as key-frames those in the middle of the video shot to reduce temporal redundancy and computation cost. Yanwei *et al.* [113] used the middle frame of each video skim segment to represent this summary in a storyboard.

## Clustering

Clustering can be used to partition a large set of data into groups, minimizing intra-group variability and maximizing inter-group separation. After partitioning, all the data selected in the same cluster have similar features. The partitioning can be based on the similarities or distances between the data points where each datum represents a vector of features of a frame. These points are grouped into clusters based on feature similarity and one or more points from each cluster are selected to represent the cluster, usually the points closest to the cluster centre. The representative points of the clusters can be used as key-frames for the entire video sequence. In some studies reported in the literature, a colour histogram was used as the clustering feature, due to its good clustering robustness. Other features can also be used. For example, Ferreira *et al.* in [91] used temporal and depth features with a clustering algorithm to segment 3D video sequences into 3D video shots.

K-means is one of the possible algorithms commonly used to solve the clustering problem. This clustering algorithm can be applied to extract key-frames from short video sequences or shots, but its application to longer video sequences must be done with care taking into

account the large processing time and memory requirements. To reduce the number of frames used by the clustering algorithm some authors pre-sample the original video, as proposed in [114]. The quality of the summaries may not be affected by this operation but the sampling rate must be chosen carefully. Although K-means is a popular and well-known clustering algorithm it has some limitations such as the need to pre-establish the number of clusters and the fact that the sequential order of the key-frames may not be preserved. Huang *et al.* [110] used the K-means clustering algorithm for extracting a set of 3D key-frames to be compared with the output of their key-frame extraction method.

### Curve simplification

In the curve simplification method each frame of the video sequence can be treated as a point in multidimensional feature space. The points are then connected in sequential order through an interpolating trajectory curve. The method then searches for a set of points which best represent the curve shape. Binary curve splitting algorithm [115] and discrete contour evolution [116, 117] are two curve simplification algorithms used in the key-frame extraction methods. Curve simplification-based algorithms preserve the sequential information of the video sequence during key-frame extraction, however the search for the best representation curve has high computational complexity. The curve simplification method proposed in [118] was also used by Huang *et al.* [110] in the evaluation process of the 3D key-frame extraction method they proposed.

### Minimum correlation

This algorithm extracts a set key-frames such that the inter-key-frame correlation is minimal, i.e., it extracts the key-frames that are most dissimilar to each other's. The optimal key-frame extraction based on minimum correlation can be defined as:

$$K = \arg \min_{l_i} Corr(f_{l_0}, f_{l_1}, \dots, f_{l_{n-1}}) \quad (2.15)$$

where  $Corr(\cdot)$  is a correlation measure,  $l_i$  is the extracted key-frame and  $K$  is a set of key-frames with  $m$  frames i.e.,  $K = \{f_{l_0}, f_{l_1}, \dots, f_{l_{m-1}}\}$ . Different algorithms can be used to find the optimal solution, such as logarithmic and stochastic search or the genetic algorithm [89]. A method for key-frame extraction from stereoscopic video, based on minimum correlation was first presented by Doulamis *et al.* in [99], using a combination of colour and depth information to summarise stereo video sequences. After segmentation

of the entire video sequence, a shot feature vector is constructed based on size, location, colour and depth for each shot. To limit the number of shot candidates, a shot selection method based on similarity between shots is applied. Finally, the stereo key-frames are extracted from each of the most representative shots. The extraction is based on the cross correlation criterion and uses a genetic algorithm [119].

### Minimum reconstruction error

In MRE based methods, the extraction of key-frames is based on minimisation of the difference between the original video sequence/shot and the sequence reconstructed from the key-frames. A frame interpolation function  $\mathcal{I}(t, K)$  is used to compute the frame at time  $t$ , of the reconstructed sequence, from a set of candidate key-frames  $K$ . The frame-copy method can be used to reconstruct the video sequence/shot (i.e., performing zero-order interpolation), but more sophisticated methods like motion-compensated interpolation might be used as proposed in [120]. The reconstruction error  $\mathcal{E}(\mathbf{F}, K)$  is defined as,

$$\mathcal{E}(\mathbf{F}, K) = \frac{1}{n} \sum_{i=0}^{n-1} d(f_i, \mathcal{I}(i, K)) \quad (2.16)$$

where  $d(\cdot)$  is the difference between two frames,  $\mathbf{F}$  is video sequence/shot with  $n$  frames,  $\mathbf{F} = \{f_0, f_1, \dots, f_{n-1}\}$ , where  $f_i$  is the  $i$ -th frame.

The key-frame ratio  $R(K)$  defines the ratio between the number of frames in the set  $K$ ,  $m$  and the total number of frames in video sequence/shot  $\mathbf{F}$ ,  $n$ , i.e.,  $R(K) = m/n$ . Given a key-frame ratio constraint  $R_m$ , the optimum set of key-frames  $K^*$  is the one that minimises the reconstruction error, i.e.,

$$K^* = \arg \min_{K \in \mathbf{F}} \mathcal{E}(\mathbf{F}, K) \quad s.t. R(K) \leq R_m \quad (2.17)$$

Thus, the MRE is defined by:

$$MRE = \mathcal{E}(\mathbf{F}, K^*) \quad (2.18)$$

For example, given a shot  $\mathbf{F}$  with  $n = 10$  frames and a key-frame ratio  $R(K) = 0.2$ , this algorithm extracts at most 2 frames as key-frames, i.e.,  $m = 2$ .

Xu *et al.* in [121] presented a key-frame extraction method to summarise sequences of 3D mesh models, wherein the number and location of key-frames are found through a R-D

optimisation process. As in all shot-based methods, shot detection is performed before key-frame extraction. In this case the SBD is based on the motion activity of a human body in dancing and sports videos. The motion activity is measured by the Euclidean distance between feature vectors of neighbouring 3D frames. The feature vectors are derived from three histograms (one for each spherical coordinate  $r$ ,  $\theta$  and  $\phi$ ) of all vertices of the 3D frames. Before the computation of spherical histograms, the Cartesian coordinates of vertices are transformed to the spherical coordinates. One of the three histograms is computed by splitting the range of the data in equal size bins. Then, the number of points from the data set that fall into each bin is counted. After shot detection, the key-frames are extracted in each shot. The key-frame extraction method is based on a R-D trade-off expressed by a Lagrangian cost function,  $cost(Shot_k) = Distortion(Shot_k) + \lambda Rate(Shot_k)$  where  $Rate$  is the number of key-frames in a shot and  $Distortion$  is the Euclidean distance between feature vectors.

Huang *et al.* [122] also presented a key-frame extraction method for 3D video based on R-D optimisation, where  $Rate$  and  $Distortion$  definitions are similar to those used in [121]. However this method is not based on shot identification, since it produces 3D key-frame summaries without requiring prior video shot detection. The key-frame summary sought should minimise a Conciseness cost function, which is a weighted sum of the  $Rate$  and  $Distortion$  functions defined in the work. A graph-based method for extracting the key-frames is used, such that the key-frames selection is based on the shortest path in the graph that is constructed from a self-similarity map. The spherical histogram of the 3D frames is used to compute the self-similarity map.

## Matrix factorisation

Another class of methods use matrix factorisation techniques to extract frames from a video sequence. Matrix Factorisation (MF) techniques are based on approximating a high dimension matrix  $\mathbf{A}$  (original data) by a product of two or more lower dimension matrices. The  $\mathbf{A}$  matrix can be composed of different features of the video or image, e.g., Gong and Liu [123] used the colour histograms to represent video frames while, Cooper *et al.* [124] computed the MF of the similarity matrix into essential structural components (lower dimension matrices). In addition to dimension reduction, the MF techniques allow reducing significantly the processing time and memory used during the operation. The MF techniques found in these key-frame extraction methods include Singular Value Decomposition (SVD) and Non-negative matrix factorisation.



Gong and Liu [123] proposed a key-frame extraction method based on SVD. To reduce the number of frames to be computed before the SVD, only a subset is taken from the input video at a pre-defined sample rate. Then, colour histograms (RGB) are used to create a frame-feature matrix  $\mathbf{A}$  of the pre-selected frames. Next, the SVD is performed on matrix  $\mathbf{A}$  to obtain an orthonormal matrix  $\mathbf{V}$  in which each column vector represents one frame in the defined feature space. Then a set of key-frames are identified by clustering the projected coefficients. According to user's request, the output can be a set of key-frames (one of each cluster) or a video skim with a user specified time duration. To construct the set of key-frames, the frames that are closest to the centres of the clusters are selected as key-frames. Non-negative similarity matrix factorisation based on low-order discrete cosine transforms [124] and sliding-window SVD [125] are other approaches for key-frame extraction based on matrix factorisation.

In [110], Huang *et al.* proposed a method to be used with 3D video to represent an animation sequence with a set of key-frames. Given an animation sequence with  $n$  frames and  $m$  vertices of a surface in each frame, an  $n \times m$  matrix  $\mathbf{A}$  is built with the vertices coordinates. This matrix  $\mathbf{A}$  is then approximately factorized into a weight  $n \times k$  matrix  $\mathbf{W}$  and a key-frame  $k \times m$  matrix  $\mathbf{H}$ , where  $k$  is the predefined number of key-frames. As  $k$  is selected to be smaller than  $n$  and  $m$ , this decomposition results in compact version of the original data  $\mathbf{A} \approx \mathbf{WH}$ . An iterative least square minimisation procedure is used to compute the weights and extract the key-frames. This procedure is driven by user-defined parameters such as a number of key-frames and an error threshold. Lee *et al.* [126] introduced a deformation-driven genetic algorithm to search good representative animation key-frames. Once the key-frames are extracted, similar to [110], the animation is reconstructed by a linear combination of the extracted key-frames for better approximation. To evaluate the performance of the proposed method, the authors compare it with Huang's method proposed in [110].

### Other methods

Other methods that cannot be classified into the preceding categories, follow different approaches. Assa *et al.* [127] proposed a method to create an action synopsis image composed of key poses (human body motion) based on the analysis through motion curve. The method integrates several key-frames into a single still image or a small number of images to illustrate the action. Currently, it is applied in 3D animation sequences and 2D video.

Lee *et al.* [8] proposed a method to select key-frames from 3D animation video using the depth information of the animation. The extracted key-frames are used to compose a single image summary. The entire video sequence is divided into temporal segments based on the motion of the slowest moving objects, and then a summarisation method is applied to the segments. The depth information and the respective gradient (computed with depth values of each frame) is used to compute the importance of each frame. A single image summary composed of several foreground visual objects is built based on the importance of each frame. The authors proposed a threshold-based approach to control the visual complexity (number of foreground objects) of the single image summary (one for each video sequence), as it is shown in Figure 2.18. By using this approach, the number of video frames to be analysed is reduced, but in some cases the method can miss important information contained in the temporal segments.

Jin *et al.* [111] proposed a key-frame extraction method for animation sequences (skeletal and mesh animations). The method uses animation saliency computed from the original data to aid the selection of the key-frames that can be used to reconstruct the original animation with smaller error. Usually, an animation sequence is characterized by a large amount of information. For computational efficiency, the animation sequence is projected to a lower-dimensional space where all frames of the sequence are represented as points of curves defined in the new lower-dimensional space. Then, the curves in the lower-dimensional space are sampled and these sampled points are used to compute the Gaussian curvature values. Next, the points with the largest curvature value are selected as candidate key-frames. Finally, a key-frame refinement method is employed to minimise an error function which incorporates visual saliency information. The aim of a visual saliency is to identify the regions of an image which attract higher human visual attention. Lee Huang *et al.* [128] expanded this idea to 3D video and computed mesh saliency for use in a mesh simplification algorithm that preserves much information of the original input. More recently, visual saliency has also been used in 3D key-frame extraction, in the method proposed by Ferreira *et al.* in [112].

Yanwei *et al.* [113] proposed a multiview summarisation method for non-synchronised views, including four of them covering 360 degrees, which results in small inter-view correlation, thus more difficult to compute similarity measures. In this method each view is segmented into video shots and general solution combines features of different shots and uses a graph model for the correlations between shots. Due to the correlation among multi-view shots, the graph has complicated connectivity, which makes summarisation

very challenging. For that purpose, Random Walks are used to do shot clustering and then the final summary is generated by a multi-objective optimisation process based on different user requirements, such as the number of shots, summary length and information coverage. The output of Yanwei's method is a multiview storyboard, condensing spatial and temporal information.

## Discussion

The problem of key-frame extraction for 3D video has been presented first by Doulamis *et al.* in [99] who proposed a method combining colour and depth information to summarise stereo video sequences. Papachristou *et al.* in [101] developed a video shot classification framework for stereoscopic video, in which the key-frame extraction method used is based on mutual information. Even though the framework was proposed for stereoscopic video, the key-frame extraction method only uses one view of the stereoscopic video. Until now, only some specific 3D video formats were considered by the existing key-frame extraction methods. Stereoscopic video was used in [99, 112], V+D is used by Ferreira *et al.* in [112] and 3D computer graphics format in [8, 110, 121, 122, 126, 127]. Thus, further research is necessary to devise efficient key-frame extraction methods for more recent 3D video formats, such as MVV, MVD and holoscopic video.

Most 3D key-frame extraction methods cited above were developed for specific content and only four of them include comparisons with similar methods [110, 111, 122, 126]. In [110], curve simplification, UnS and clustering methods were utilized as reference for performance evaluation and comparison of the proposed matrix factorisation methods. The authors showed that the method based on matrix factorisation extracts more representative key-frames in comparison with the other three competing methods [111, 122, 126]. However, the algorithm is very slow with quadratic running time complexity. In [126], the proposed method based on genetic algorithm is compared with Huang's method [110] in terms of the Peak Signal-to-Noise Ratio (PSNR) and computational complexity. The former is very efficient in terms of computation time when compared to the latter but qualitywise (average PSNR) it is slightly worse. However, Huang's method [110] is slightly better when comparing maximum and minimum PSNR.

Peng Huang *et al.* in [122] confront their key-frame extraction method with the method used in [121] and the results show improved performance for all 3D video sequences used in tests. Jin *et al.* in [111] also compare the proposed method with the UnS and Principal

Component Analysis methods [129]. The results show that the proposed method achieves much better reconstruction of skeletal and mesh animation than the other methods under analysis.

As mentioned before, most of the key-frame extraction methods for 3D video, rely on a previous SBD step. However, the methods just described, from [110, 111, 122, 126], do not perform any pre-analysis of the video signal to identify shots and their boundaries. Therefore, the quality of key-frame summaries obtained by using such approach can be negatively affected when accurate shot segmentation is not available. Another important issue is the definition of the number of key-frames that is needed to represent the original sequence. This number depends on user requirements and on the content of the video to be summarised and its choice frequently involves a trade-off between the quality and efficiency of the key-frame summary.

### 2.4.3 Presentation of video summaries

Once the key-frames are extracted to form a video summary, they need to be presented to users in some organized manner to facilitate video browsing and navigation operations. The video summary presentation methods aim to show the key-frames in some meaningful way, allowing the user to grasp the content of a video without watching it from beginning to end [89]. The most common methods for key-frame presentation are the static storyboard, dynamic slideshow and single image.

Static storyboard presents a set of miniaturised key-frames spatially tiled in chronological order, allowing a quick browsing and viewing of the original video sequence. This presentation method was used with 3D video in [99, 110, 111, 121, 122, 126]. The second method is the dynamic slideshow, that presents the key-frames one by one on the screen, which allows browsing over the whole video sequence. Another presentation method is the single image, which morphs parts of different key-frames in chronological order to produce a single image. Normally, in this presentation type the background and foreground objects (time shifted) are aggregated in single image, as exemplified in Figure 2.16. In this figure, the foreground is the children who plays in the bars of a playground. Here it is presented, three positions of the children in the bars, which correspond to three key-frames of video sequence.

Qing *et al.* [130] proposed a generic method for extracting key-frames in which the Jensen-Shannon divergence is used to measure the difference between video frames to segment



Figure 2.16: Video synopsis proposed [6].

the video into shots and to choose key-frames in each shot. The authors also proposed a 3D visualisation tool, to display key-frames and other useful information related to the key-frame selection process. More recently, Nguyen *et al.* [131] proposed the Video Summagator, which provides a 3D visualisation of a video cube of static and dynamic video summaries. Assa *et al.* [127] proposed a method to create an action synopsis image from a 3D animation sequence or 2D video. Lee *et al.* [8] also proposed a method to summarise a 3D animation into a single image based on depth information.

In [7] a 3D interface (3D-Ring and 3D-Globe) was proposed as an alternative to the 2D grid presentation for interactive item-search in visual content databases, (see Figure 2.17). Even though this system was designed to be used with a large database, it can also be applied to visualize key-frames summaries of 2D and 3D video.

## Discussion

Most of the 3D key-frame extraction methods proposed in the literature until now are focused on the extraction rather than in the presentation of key-frame sets to the viewers. So far only Assa *et al.* and Lee *et al.* proposed in [127] and [8] two presentation solutions distinct from the static storyboard used in association with most of 3D key-frame extraction methods [99, 110, 111, 121, 122, 126]. In this scenario, with only two presentation solutions, it is foreseeable that the development of new 3D video and image display devices will lead to the creation of new methods to display 3D video summaries or key-frame collages providing the user with more immersive and more meaningful ways to observe these types of time-condensed video representations.

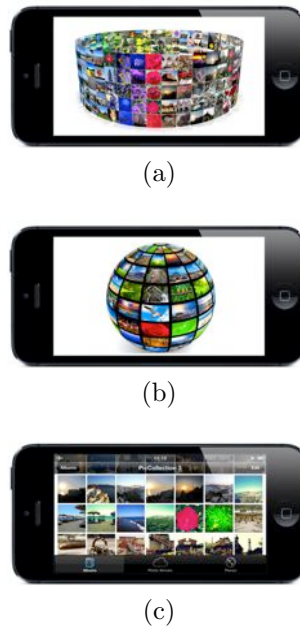


Figure 2.17: (a) 3D-Ring interface, (b) 3D-Globe interface and (c) 2D grid presentation (figure based on [7]).



Figure 2.18: Single image key-frame presentation method [8].

#### 2.4.4 Performance metrics

One of the most important topics in the development and validation video summarisation algorithms is the performance evaluation. In this section is presented the key-frame summary evaluation methods and some related aspects. These methods are classified in three groups: result description, subjective and objective methods, as it was proposed in [89].

## Result description

The result description is the most common form of performance evaluation, since it does not require a reference, either for objective or subjective comparison with other methods. Usually, it is used to explain and describe the advantages of some method in comparison with others based on presentation or/and description of the key-frames extracted (visual comparison), as in [8, 110, 111, 121, 122, 126]. This type of evaluation can also be used to discuss the influence of specific parameters or features of the method and also the influence of the content in the key-frame set, as in [99, 121]. In some works, this type of evaluation method is complemented with objective and/or subjective methods as in [8, 121]. However the Result Description method has some limitations, such as the reduced number of methods which can be compared at same time i.e., it is inadequate to compare key-frame summaries of a large number of video sequences or methods. Another drawback is the subjectivity inherent to this type of evaluation, since the underlying comparisons results are usually user-dependent and so prone to inter and intra observer fluctuations.

## Subjective methods

Subjective methods rely on the independent opinion of a panel of users judging the quality of the generated key-frame video summaries according to a known methodology and criteria. In this type evaluation, a panel of viewers are asked to observe both the summaries and the original sequence and then respond to questions related to some evaluation criteria, (e.g., “Was the key-frame summary useful?”, “Was the key-frame summary coherent?”) or if each key-frame is “good”, “fair”, or “poor” according to the original video sequence.

The experiments can include a set of absolute evaluations and/or a set of relative evaluations, in which two key-frame summaries are presented and compared. Usually, the summary visualisation and rating steps are repeated for each video in the evaluation set by each viewer. During the evaluation of the key-frame summaries it is also required taking into account the external factors which can influence the ratings of the summaries, such as the attention and fatigue specially when there are long evaluation sessions with many video summaries. In addition to these factors, the experiments should follow standard recommended protocols prepared specifically for subjective assessment of video quality [132].

Subjective evaluation methods were used in [59, 63, 64, 85, 113]. In [59], subjective assessment was used to grade the single key-frame representations as “good”, “bad” or “neutral” for each video shot and also to provide appreciations about the number of key-frames with possible grades being “good”, “too many”, “too few” in the case of multiple key-frames per shot. In [64, 85], the quality of the key-frame summary is evaluated by asking users to give a mark between 0 to 100 for three criteria, “informativeness”, “enjoyability” and “rank” after watching the original sequences and the respective key-frames summaries. Ejaz *et al.* [63] used subjective evaluations to compare the proposed method with four prominent key-frame extraction methods: Open Video project (OV) [115], Delaunay Triangulation (DT) [133], STill and MOving Video Storyboard (STIMO) [134] and Video SUMMarisation (VSUMM) [114]. In this case, the evaluation is based on Mean Opinion Scores (MOS) and viewers are asked to rate the quality of the key-frame summary using a scale from 0 to 5 after watching the original sequences and the respective summaries generated by all methods.

In [113] subjective assessments were also used to evaluate multi-view video summaries. The aim is to grade the “enjoyability”, “informativeness” and “usefulness” of the video summary. Here, three questions were asked to viewers to evaluate the method:  $Q_1$ : “How about the enjoyability of the video summary?”  $Q_2$ : “Do you think the information encoded in the summary is reliable compared to the original multi-view videos” and  $Q_3$ : “Will you prefer the summary to original multi-view videos if stored in your computer?”. In reply to the questions  $Q_1$  and  $Q_2$  the viewers assigned a score between 0 to 5 and for  $Q_3$  the viewers only need to respond with “yes” or “no”. From all 3D key-frame extraction methods reviewed, only those presented in [8, 113, 127] used subjective evaluations.

## Objective methods

Although subjective evaluation provides a better representation of the human perception than objective methods, it is not suitable for practical implementations due to the time and number of users required to obtain valid scores. Objective evaluation methods are reproducible and can be specified analytically. Since they are automatable can be used to rate the proposed method on large number of videos of variable genres and formats. These methods can be applied to all types of video formats without requiring the involvement of human observers and can be performed rapidly and automatically by using objective quality metrics.



The methods reviewed in this section use objective quality evaluation and employ several quality measures originally developed for 2D video, but can be also applied to 3D video, after being modified to take into account the specific features of 3D visual information. The Shot Reconstruction Degree (SRD) distortion measure [135] and the Fidelity measure ( $Fm$ ) defined in [136] follow two different approaches. Fidelity measure employs a global strategy, while SRD uses a local evaluation of the key-frames. To judge the conciseness of a key-frame summary a measure of the Compression Ratio (CR) is used [137]. If a ground-truth summary is available the Comparison of User Summaries (CUS) [114], Recall rate, Precision rate and accuracy measure (F1) measures can be used. These measures compare the computed summaries with those manually built by users. More details on these measures are presented in the next sub-sections.

### Shot reconstruction degree

SRD measures the capability of a set of key-frames to represent the original video sequence/shot. Assuming a video shot  $\mathbf{F} = \{f_0, f_1, \dots, f_{n-1}\}$  of  $n$  frames and  $K = \{f_{l_0}, f_{l_1}, \dots, f_{l_{m-1}}\}$  a set of  $m$  key-frames selected from  $\mathbf{F}$ , the reconstructed scene shot  $F' = \{f'_0, f'_1, \dots, f'_{n-1}\}$  is obtained from the  $K$  set by using some type of frame interpolation. The SRD measure is defined as:

$$SRD(\mathbf{F}, F') = \frac{1}{n} \sum_{k=0}^{n-1} Sim(f_k, f'_k) \quad (2.19)$$

where  $n$  is the size of the original video sequence/shot  $\mathbf{F}$  and  $Sim(\cdot)$  is the similarity between two video frames. In Liu *et al.* [135], the similarity measure chosen was PSNR, but other similarity metrics that include 3D features can also be used in the evaluation of 3D key-frame summaries. A  $K$  key-frame summary is a good representation of the original  $\mathbf{F}$  when the magnitude of its SRD is high.

### Fidelity

The Fidelity,  $Fm$  is computed as the maximum of the minimal distances between the set of key-frames  $K$  and each frame of the original  $\mathbf{F}$ , i.e., a Semi-Hausdorff distance  $d_{sh}$ . Let  $\mathbf{F}$  be a video sequence/shot containing  $n$  frames, and the set  $K = \{f_{l_0}, f_{l_1}, \dots, f_{l_{m-1}}\}$  of  $m$  frames, selected from  $\mathbf{F}$ . The distance between the set  $K$  and a generic frame  $f_k$  s.t.

$0 \leq k \leq n - 1$  belonging to  $\mathbf{F}$  can be calculated as follows.

$$d_{min}(f_k, K) = \min_j \{d(f_k, f_{l_j})\}; j = 0, 1, \dots, m - 1 \quad (2.20)$$

Then the semi-Hausdorff distance  $d_{sh}$  between  $K$  and  $\mathbf{F}$  is defined as:

$$d_{sh}(\mathbf{F}, K) = \max_k \{d_{min}(f_k, K)\}; k = 0, 1, \dots, n - 1 \quad (2.21)$$

The Fidelity measure is defined as:

$$Fm(\mathbf{F}, K) = MaxDiff - d_{sh}(\mathbf{F}, K) \quad (2.22)$$

where *MaxDiff* is the largest possible value that the frame difference measure can assume. The function  $d(f_a, f_b)$  measures the difference between two video frames  $a$  and  $b$ . The majority of the existing dissimilarity measures can be used for  $d(\cdot, \cdot)$ , such as the  $L_1$ -norm (City block distance),  $L_2$ -norm (Euclidean distance) and  $L_n$ -norm [136]. As it was mentioned before, the *Fm* measure can be used for 3D video with the necessary changes in the  $d(\cdot, \cdot)$  distance. Whenever *Fm* is high, this means that the selected key-frames provide an accurate representation of the whole  $\mathbf{F}$ .

### Compression ratio

A video summary should not contain too many key-frames since the aim of the summarisation process is to allow viewers to quickly grasp the content of a video sequence. For this reason it is important to quantify the conciseness of the key-frame summary. The conciseness is the length of the key-frame video summary in relation to the original video segment length and can be measured as a compression ratio, defined as the relative amount of “savings” provided by the summary representation:

$$CR(\mathbf{F}) = 1 - \frac{m}{n} \quad (2.23)$$

where  $m$  and  $n$  are the number of frames in the key-frame set  $K$  and the original video sequence  $\mathbf{F}$  respectively. Generally, high compression ratio is desirable for a compact video summary [137].

### Comparison of User Summaries

CUS is a quantitative measure based on the comparison of summaries built manually by users and computed summaries. It was proposed by Avila *et al.* in [114]. The user summaries are taken as reference, i.e., the ground-truth, and the comparison between the summaries is based on specific metrics. The colour histogram is used for comparing key-frames from different video summaries, while the distance between them is measured using the Manhattan distance. Two key-frames are similar if the Manhattan distance of their colour histograms is below than a predetermined threshold  $\delta$ . In [114], this threshold value was set to 0.5. Two evaluation metrics, accuracy rate  $CUS_A$  and error rate  $CUS_E$ , are used to measure the quality of the computed summaries. They are defined as follows:

$$CUS_A = \frac{n_{match}}{n_{US}} \quad CUS_E = \frac{n_{no-match}}{n_{US}} \quad (2.24)$$

where  $n_{match}$  and  $n_{no-match}$  are, respectively, the number of matching and non-matching key-frames between the computed and the user generated summary and  $n_{US}$  is the total number of key-frames in the summary.  $CUS_A$  varies between 0 and 1, where  $CUS_A = 0$  is the worst value indicating that none of the key-frames from the computed summary matches those of the user summary. A value of  $CUS_A = 1$  is the best case and indicates that all key-frames from both summaries perfectly match each other. A null value for  $CUS_E$  indicates a perfect match between both summaries.

### Computational complexity

Another relevant performance metric taken into account in the evaluation of key-frame extraction methods is the computational complexity, which is usually equated with the time spent to construct a key-frame summary. This metric was used in [63, 64, 114, 137, 138] for 2D video summaries. In 3D key-frame extraction methods, the computational complexity metric is only used by Lee *et al.* in [126], where the computational complexity of Lee's and Huang's *et al.* [110] methods are compared.

### Other methods

Other methods and measures were used for objective evaluation of 3D key-frames summaries. In [121, 122] a rate-distortion curve is used, modelling a monotonic relationship

between rate and distortion, with increases of the former leading to decreases of the latter. In the work described in [110], the Root Mean Square Error (RMSE) distance between the original and reconstructed animation was used as the objective quality measure (with an inverse relationship in this case). This measurement is the same as in [139] and [140]. Lee *et al.* [126] used PSNR to measure the reconstruction distortion. Jin *et al.* in [111] measure reconstruction error of the animation from the extracted key-frames, using average of Degrees of Freedom (DOF) of reconstruction error magnitude.

## Discussion

Conciseness, coverage, context and coherence are desirable attributes in any key-frame summary as a flexible representation of video sequences. Some of these attributes are mostly subjective, such as the context and coherence. Conciseness is related to the length of the key-summary, while the coverage evaluation is based on comparison between computed key-frames summary and ground-truth summary, expressed by the Recall rate, Precision rate,  $CUS_A$  and  $CUS_E$ .

Most evaluation metrics reviewed above were developed for 2D video. However, some of them, such as  $Fm$  and SRD, have also been extended to evaluate 3D video summaries after some adaptation. This is the case of the 3D key-frame extraction method presented by Ferreira *et al.* in [112], where the  $Fm$  and SRD metrics were used. To measure the Recall rate, Precision rate,  $CUS_A$ ,  $CUS_E$ , computational complexity and compression ratio in 3D video summarisation, no adaptation is needed.

The key-frame extraction methods are often application-dependent (e.g., summarisation of sports videos, news, home movies, entertainment videos and more recently for 3D animation) and the evaluation metrics must be adapted to the intended use. A good summary quality evaluation framework must be based on a hybrid evaluation scheme which includes the strengths of subjective and objective methods and also the advantages of result description evaluations.

### 2.4.5 Applications

In this section some applications of 3D summarisation and some aspects related to these applications were presented. These applications are grouped in five categories: video browsing, video retrieval, content description, animation synthesis and others.

### **Video browsing**

The video browsing and associated problem has been investigated by the research community for decades, [141]. However, the growing use of 3D video and the specific characteristics of this type of visual information make 3D video browsing a more interesting and challenging problem. The access to databases or other collection of videos could be eased by the use the key-frame extraction methods to abstract/resume long video sequences in the repository of interest. With this kind of abridged video representation, a viewer can quickly find the desired video in a large database. For example, once an interesting topic has been identified through display of the key-frames, a simple operation as a click on the respective key-frame can initiate video playback of the original content at that particular instant. Many video browsing methods have been proposed for 2D video [141]. However, to the best of the authors' knowledge, in the case of 3D video there are no works reported in the literature.

### **Video retrieval**

In contrast to video browsing, where viewers often just browse interactively through video summaries in order to explore their content, in video retrieval the viewers search for certain visual objects (e.g., objects, people and scenes) in a video database. In this type of retrieval processes, viewers are typically expected to know exactly what they are looking for. Therefore, it is crucial to implement appropriate search mechanisms for different types of queries provided by distinct viewers and with particular interests. Matching the viewers' interests (queries) with the database content can be made with recourse to textual, image based descriptions or combinations of both. Some 2D video search and retrieval applications have combined video browsing and retrieval in the same framework [141]. In the case of 3D video this problem is still open for research, as no similar solutions exist in the available literature. Finally, it is worth to point out that work done on 3D object recognition techniques which can also be used in retrieval, as published in [142–144].

### **Content description**

Vertos *et al.* [145] presented a way of using the Audio-Visual Description Profile (AVDP) profile of the MPEG-7 standard for 3D video content description. The description of key-

frames is contemplated in the AVDP profile through the *MediaSourceDecompositionDS* (i.e., *MediaSourceDecompositionDS* is used in the AVDP context to decompose an audiovisual segment into the constituent audio and video channels). Thus, this content description scheme, allows to use 3D key-frames for fast browsing and condensed representation of query results of 3D video search tasks. Other application of key-frames to content description was proposed by Sano *et al.* [146]. Here, the authors proposed and discussed how the AVDP profile of the MPEG-7 can be applied to multi-view 3D video content [129].

### **Animation synthesis**

Blanz *et al.* [147] proposed a morphable 3D face model by transforming the shape and texture of example into a new 3D model representation. According to this modelling approach, new or similar faces and expressions can be created by forming linear combinations of the 3D face models. A similar concept to the one proposed in [147] can be applied to generate 3D models [148] or to synthesize new motion from captured motion data [149]. Animation synthesis based on key-frames [150] using the same concept has been presented in [147–149], to interpolate frames between two key-frames. However, the quality of the interpolated frames is dependent on the inter-key-frame distance and on the interpolation method used.

### **Other applications**

Assa *et al.* [127] proposed the use of action synopsis images as icons (personal computer desktop and folders) and thumbnails of the 3D animation. Assa *et al.* also proposed an automatic or semi-automatic generation method to create comic strips and storyboards for 3D animation. Lee *et al.* [8] presented a method to create a single image summary of a 2D or 3D animation, which can be used in the same application as Assa's work. Halit *et al.* [129] proposed a tool for thumbnail generation from motion animation sequences. Several authors, as [151–154] have used key-frame extraction methods in the 2D-to-3D video conversion.

## 2.5 Flexible video coding

In general flexible video coding allows different coded representations of visual content, comprising objects/regions and frames. This is usually known as scalable coding and ROI-based coding. This section presents an overview of coding schemes that allow flexible representation of coding content. Firstly, basic concepts of the video coding are discussed, then scalable and ROI video coding are reviewed. The performance metrics used to evaluate the coding efficiency and some applications of scalable and ROI video coding are also presented.

### 2.5.1 Basic concepts

Video compression is based on eliminating redundant and irrelevant (which is not perceived by eye/brain) information of the source. Compression can be lossless (reduction of redundant information without loss of any information) or lossy (with lost information). Lossy compression is normally used in image and video coding by most video compression standards based on intra and inter-frame coding. Spatial redundancy is due to correlation between pixels in the same image/frame. If correlation exists in the spatial domain (i.e., neighbouring pixels have similar values), redundancy can be reduced through intra-frame prediction. In the case of temporal domain, the temporal redundancy is due to similarities between adjacent or near frames.

The main coding tools used in hybrid video coding are inter and intra-frame prediction, transform, quantisation and entropy coding. The prediction unit is usually followed by the transform and quantisation of prediction residues, which is then succeeded by the entropy coding. The entropy coding is used to exploit the statistical data redundancy. In hybrid video codec each input frame is divided in blocks, in which the block size is dependent on the prediction mode used. In inter-frame prediction, each block is predicted with information used in others encoded frames, typically use motion compensation. Contrary to inter-frame prediction, in intra-frame prediction no information of other frames is used i.e., each block is predicted from the information used in neighbouring blocks. Intra-frame coding ensures that systematic errors do not continuously propagate, throughout the sequence since an entire frame is periodically encoded on its own.

Motion Estimation (ME) is used in inter-frame prediction to exploit the fact that in most video sequences the difference between two adjacent frames results from camera or object

motion. By using ME, the encoder is able to encode only the difference between two frames, discarding the redundant information between them. The motion estimation unit finds Motion Vectors (MV) for each block, i.e., motion estimation of a block involves finding  $n \times n$  region in a reference frame that closely matches the current block. The MV and previously reconstructed frame are fed to the motion compensation unit to create the inter-frame prediction.

The prediction obtained from intra and inter-frame unit is subtracted from the current block to produce a prediction residue or prediction error. Then the residue is transformed from the spatial domain to the frequency domain in order to de-correlate the signal and concentrate the energy in a few coefficients. Then, each sub-block is quantised and the small values associated to spectral components that are not perceptually relevant are eliminated. Finally, the coefficients, motion vector and associated header information for each block are entropy encoded to produce the compressed bit stream.

## 2.5.2 Scalable video coding

The concept of scalability in video coding can be viewed as the possibility of extracting part of a coded video stream in order to adapt the bit stream to heterogeneous networks, different needs or preferences of the users as well as terminal capabilities. This multi-layer representation can be made in different domains, such as the temporal, spatial, quality and in the combination of these three scalability domains. Therefore scalable video is also regarded as a flexible representation of visual content by encoding it in different layers with respect to various parameters. Figure 2.19 shows a representation of temporal, spatial and quality scalability. Next, a brief description of these scalability types will be provided.

**Spatial scalability** - produces a scalable stream with multiple spatial resolutions and also enables extracting and decoding different spatial resolutions from the scalable video stream. The coding information from the lower resolution is used for prediction of the higher resolution to increase the coding efficiency of the higher resolution. Figure 2.19 shows a scalable representation coded with tree spatial resolutions (QCIF, CIF and 4CIF), two qualities and two temporal resolutions.

**Temporal scalability** - produces a scalable stream with multiple temporal resolutions, i.e., the frames of the video sequence are divided into layers, in which the base layer



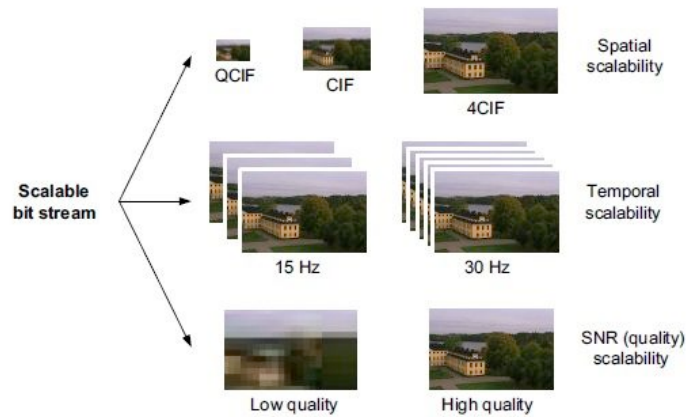


Figure 2.19: Scalability types.

is independently coded to provide the lower temporal rate and the others layers are coded with temporal prediction with respect to base layer. For instance, in Figure 2.19 the scalable includes two temporal rates (15Hz and 30Hz). Therefore different temporal resolutions can be extracted from the scalable video stream.

**Quality scalability** - produces a scalable bit stream with a single spatial and temporal resolutions but with different qualities. The coding information of the lower quality layers is used for prediction of the higher quality layers. In the Figure 2.19 it is shown a scalable bit stream with low and high quality. This type of scalability also enable to extract the video content of different qualities from a scalable bit stream.

**Combined scalability** - produces a scalable bit stream with different spatio-temporal resolutions and bit rates as the result of the combination of spatial, temporal and quality scalability.

Scalable video coding is a flexible representation particularly tailored for heterogenous communications environments where users, networks and devices are quite different and they all are granted access the same content.

### Scalable video coding - SVC

Previous video coding standards such as H.262|MPEG-2 Video, H.263 and MPEG-4 Visual also contain the scalable profiles but they have rarely been used because spatial and quality scalability features came along with a significant loss in coding efficiency as well as

a large increase in encoder complexity as compared to the corresponding single-layer profiles. Scalable extension of the H.264/MPEG-4 AVC video coding standard has produced substantial improvements in terms of coding efficiency and scalability compared to scalable profiles of the previous video coding standards without significantly increasing the decoding complexity. H.264/MPEG-4 SVC standard [11] provides the same compression functionality of the H.264/MPEG-4 AVC standard, but new coding tools for the generation of scalable bit stream were implemented. H.264/MPEG-4-SVC is based on a layered scheme, in which the bit stream is coded into base layer, H.264/MPEG-4 AVC compliant, and one or more enhancement layers, as it is shown in the block diagram of an SVC encoder of the Figure 2.20. Each video signal with specific resolution (grey area in the Figure 2.20) is coded in scalable bit stream and they are characterized by a layer identifier (layer 0 or base layer, layer 1, ..., layer  $n$ ). To exploit the dependencies between layers and to improve the coding efficiency of enhancement layers, the H.264/MPEG-4-SVC provides the inter-layer motion prediction, inter-layer residual prediction and inter-layer intra prediction. These inter-layer predictions modes are represented with dash arrows in Figure 2.20 [11].

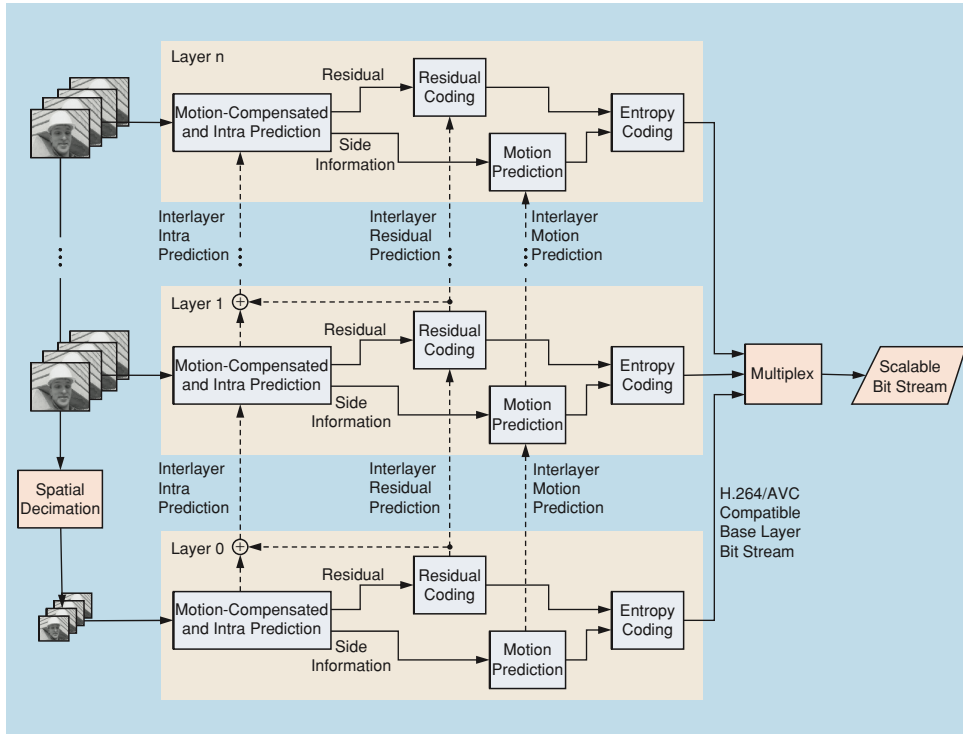


Figure 2.20: Simplified SVC encoder architecture [9].

The SVC bit stream is organised in such way that enables a user to easily extract only

a subpart of the data contained in the scalable bit stream while still being able to decode the original input video at a reduced spatial resolution, frame rate or quality. The H.264/MPEG-4 SVC supports temporal, spatial and quality scalability and each can be available with different granularity.

**Temporal scalability** - allows to code video with multiple frame rates in a single bit stream with specific syntax (scalable bit stream) and also enable to decode subparts of this scalable bit stream. For example, if the original video signal contained 30 frames per second is coded as scalable bit stream, the temporal scalability would enable a user to decode a subpart of scalable bit stream, generating a sequence with 15 or 7.5 frames per second. In H.264/MPEG-4-SVC, the temporal scalability is provided by the concept of hierarchical B-pictures, as it is shown in Figure 2.21. In this example, the prediction structure provides four temporal scalability levels (T0, T1, T2 and T3). The coding order and display order of the frames is also presented in Figure 2.21.

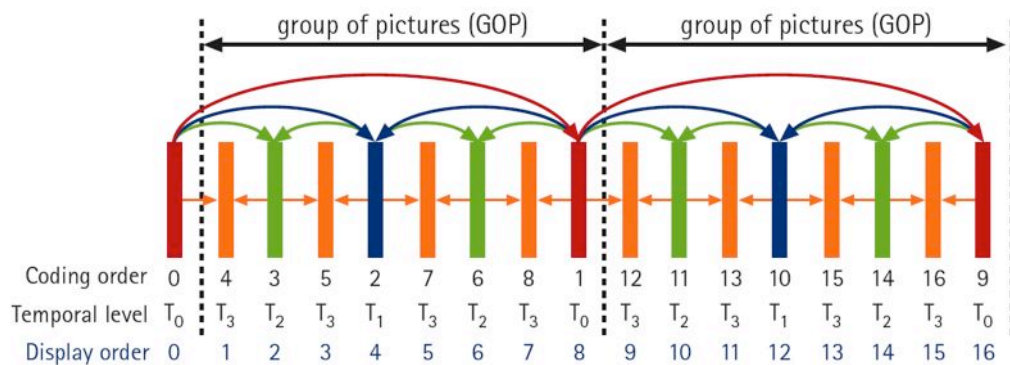


Figure 2.21: Temporal scalable architecture of SVC [10].

The hierarchical B-pictures structure uses bidirectional predictive pictures (B-pictures) as references to other B-pictures within one Group of Pictures (GOP). In general, bidirectional prediction can get more accurate prediction than unidirectional prediction due to which B-pictures are coded more efficiently than P-pictures. After coding, all frames are reordered to allow that frames of the lower layers to be extracted first. The prediction structure of the GOP must be chosen such that a frame is only predicted from a lower or the current enhancement layer, thus ensuring no dependencies from higher layers. To decode higher layer it is necessary decoding all layers below it, due to prediction dependency between these layers.

The scalable bit stream is organized in Network Abstraction Layer (NAL) units, which

are composed by payload and header with several syntax elements. Each NAL unit belongs to a particular spatial, temporal and quality layer. The information stored in header (the syntax element) of each NAL unit identifies each layer (spatial, temporal and quality). The syntax element of the temporal scalability is the temporal identifier  $T$ , where the temporal base layer is identified by the  $T = 0$  and  $T$  of the remaining temporal layers is modified by adding one unit to the previous  $T$  value, for instance in the first enhancement temporal layer  $T = 1$ . To conclude, the hierarchical B-pictures structure provides temporal scalability but also shows good coding efficiency in comparison to simulcasting [11].

**Spatial scalability** - to support the spatial scalability coding, SVC uses the multi-layer coding approach as the previous video coding standards with scalable profiles. Each spatial layer has dependency identifier  $D$ , which it is increased by 1 from spatial base layer ( $D = 0$ ) to the following layer. The resolution of base layer (H.264/MPEG-4 AVC compliance) corresponds to the low resolution video sequence. Normally, the resolution of each enhancement layer contains a higher resolution than the previous layer. In order to improve the coding efficiency of SVC in comparison to simulcasting, inter-layer prediction was incorporated in this scalability mode, as illustrated in Figure 2.22. The inter-layer prediction uses the lower layer information (MBs types, motion parameters, residual signal) in the enhancement layers coding. This scheme improves rate-distortion performance of the enhancement layers.

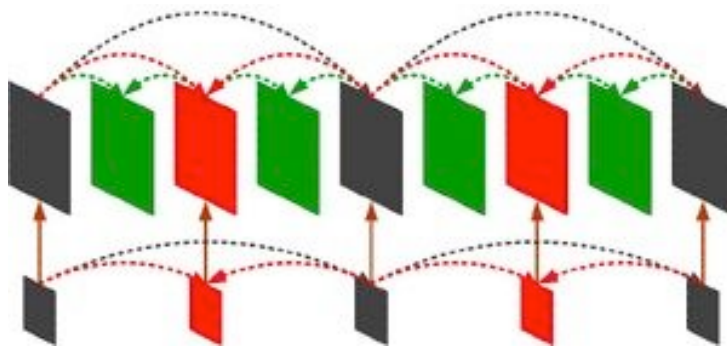


Figure 2.22: Multi-layer structure with inter-layer prediction [11].

**Quality scalability** (or Signal-to-Noise Ratio (SNR) scalability) - the H.264/MPEG-4 SVC supports two quality scalable modes, namely Coarse Grain Scalability (CGS) and Medium Grain Scalability (MGS). The CGS is similar to the spatial scalability in H.264/MPEG-4 SVC. This scalable mode employs the same inter-layer predictions

as the spatial scalability, such as prediction of MBs modes, motion parameters and prediction of the residual signal [11], however, the spatial resolution is kept constant, i.e., the base and enhancement quality layers have the same size. To increase the quality between adjacent CGS layers, the residual signal in the enhancement layer is re-quantized with a Quantisation Parameter (QP) smaller than the QP of the previous CGS layer. H.264/MPEG-4 SVC allows at most eight CGS layers which represented to eight quality extraction points, i.e., one base quality layer and up to seven enhancement layers [155]. MGS mode is more flexible than CGS, i.e., MGS mode allows a finer granularity level of quality scalability by dividing a given enhancement layer into various MGS layers [11]. In H.264/MPEG-4 SVC each MGS layer is distinguished by a quality layer index  $Q$  where  $Q = 0$  correspond to the MGS base layer. In each spatial layer  $D$  more than one MGS quality layers  $Q$  can exist up to the maximum of 16 MGS layers. The MGS split the particular enhancement layer of a given video frame into up to 16 MGS layers. In detail, the MGS splits the transform coefficients of generic frames into multiple groups and each group is assigned to a particular MGS layer.

**Combined scalability** - the combination of spatial, temporal, and SNR scalability increased the flexibility of the scalable bit stream, however the coding efficiency of this approach is slightly worse than that of the layered approach, but it provides more decodable points in the spatial, temporal and quality levels [11, 156].

In summary, the scalable coding is more flexible, offers more functionalities and scalable profiles than single-layer coding. In addition, the coding efficiency of the SVC is clearly better than the single-layer coding (simulcasting), since the single-layer coding supports all spatio-temporal resolutions and the bit rates in separated bit streams [11].

### Scalable video coding - SHVC

The Joint Collaborative Team on Video Coding (JCT-VC) of experts from the ITU-T Visual Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) are currently developing a new video coding standard with the name High Efficiency Video Coding (HEVC) [157]. This new video coding provides a bit rate reduction in range of 40%-50% for the same quality compared to H.264/MPEG-4 AVC, in single-layer coding [158]. In order to address the potential needs of future video applications

and the network limitations, the JCT-VC issued a Call for Proposals (CfP) on Scalable Video Coding Extensions for High Efficiency Video Coding.

In [159], Choi *et al.* proposed the first solution for spatial scalability to the HEVC. Here, an inter-layer prediction mechanism for single-loop and multi-loop decoding is presented. In the multi-loop decoding, only inter-layer texture is employed while in the single-loop decoding the texture prediction and inter-layer motion prediction with inter-layer residual prediction are used. The enhancement inter-layer motion and residual prediction is determined by the up-sampling operation of motion vectors and residual information from base layer respectively. This proposal provides a bit rate reduction of up to 10% compared with HEVC simulcast for the single-loop and multi-loop decoding for the all-intra (JCT-VC test configuration). For the Random Access test the bit reduction is up to 7.4% and 2.6% for the multi-loop and single-loop decoding respectively, compared with HEVC simulcast.

Shi *et al.* proposed in [160] spatially scalable video coding for HEVC. This proposal supports single-loop and multi-loop solutions as the Choi *et al.* proposal [159]. Here, two inter layer prediction mechanisms are used to reduce redundancy between layers, the basic Q-mode and the extra L-mode. The basic Q-mode includes intra, motion vectors with code modes and residuals. An extra path learning prediction mode is proposed to improve the accuracy of the inter layer prediction, here the authors defined this new mode as extra L-mode. Experimental results demonstrate the effectiveness of the proposed scheme compared with HEVC simulcast. Since the temporal scalability is already HEVC supported by the flexible reference picture, the challenges of the scalable HEVC extension are the spatial and quality scalability [161]. Several proposals for the HEVC scalable extension were previously presented, some of them used inter layer prediction based on the SVC standard.

Hinz *et al.* [161] present a review of scalable HEVC extension with spatial and quality scalability. In this paper a review of the scalable coding tools (i.e., inter layer prediction, entropy coding transform coefficients) used in the HECV and simulation tests for multi-loop decoding are made. The HEVC scalable coding tools are compared with scalable coding tools of scalable extension of the H.264/MPEG-4 AVC. The results show the increase in coding efficiency from HEVC relatively to the SVC coding tools.

### 2.5.3 ROI based video coding

ROI may be computed through visual attention models, as described before. ROI coding aims at providing better quality and/or better protection against the errors in the ROI as opposed to the rest of the visual scene. This differentiated coding can be used in various types of video applications where some kind of perceptual coding may be useful or to increase the coding efficiency in constrained video delivery systems. Some coding systems allow assigning higher priority and quality to ROI over the rest of the frame (non-ROI). The quality and priority difference between the ROI and non-ROI will depend on the video application. For example in video communications the use of the available bit rate should be maximized to provide the best quality to viewers. In this case lower quality non-ROI is acceptable by the human visual system since the viewers pay less attention to the non-ROI than to the ROI. Therefore, more bits can be allocated to the ROI without reducing the overall quality of the video. In this case, the ROI quality can be increased. In the literature, there are different approaches to achieve this goal: (i) use more bits in ROI (use a fine QP); (ii) reduce the bits in the non-ROI (use a great QP) and (iii) code the non-ROI regions in skip-mode. In addition to these approaches, several works proposed to adjust the QP of the ROI and non-ROI according to certain principles, such as the visual sensitivity of the human visual system or the ROI quality target. Chen *et al.* proposed a method that minimizes the non-ROI information based on application of low-pass filters to the non-ROI region. In this case it is not necessary to do any modification to the coder, since the technique is applied before the coding [162]. The same approach was used by Karlsson *et al.*, here, a spatio-temporal filter to re-allocate bits from the non-ROI to the ROI, after the ROI detection step was proposed [163].

#### ROI coding with H.264/MPEG-4 AVC

In H.264/MPEG-4 AVC the MBs are grouped into an entity called a slice. Each slice is identified with slice group ID. The slice group ID determines the coding order of the MBs, for instance, in the case of two slice groups, all the MBs of slice group 0 are coded before the coding the MBs of slice group 1. In the case of H.264/MPEG-4 AVC the maximum number of slice group for each frame is 8, i.e., each frame can only be split into 8 slices group. Conventionally, video coding standards support encoding MBs only in raster order. However, with the introduction of Flexible Macroblocks Ordering (FMO) in H.264/MPEG-4 AVC standard, the order of assigning the MBs into slices has been

liberalized. Here, the slides are coded into separate NAL units [164] making them totally autonomous from others. The H.264/MPEG-4 AVC standard defines seven different types of FMO modes: Interleaved, Dispersed, Fore-ground with left-over, Box-out, Raster-scan, Wipe, and Explicit. The FMO type 2 has usually been used in ROI coding, but it is only suitable for ROIs with rectangular shapes and cannot represent irregular regions in an efficient manner. In this case, the type 6 is the most general one, here the ROI shape is user-defined. The H.264/MPEG-4 AVC standard did not have a tool for ROI detection. Thus, the ROI position must be first detected with a pre-processing algorithm.

Leuven *et al.* [165] present an implementation of multiple ROI models in H.264/MPEG-4 AVC standard to enhance the quality of video surveillance. The ROIs are user-defined, i.e., no detection algorithm is used. The results show that a convenient selection of the ROI, in combination with a suitable choice of quantisation parameter and the FMO type can reduce bandwidth usage while maintaining the same video quality. More recently, Peng *et al.* [166] present a ROI privacy protection scheme for H.264/MPEG-4 AVC video in Closed Circuit TeleVision (CCTV) based on FMO. To encrypt the ROI the FMO technology of the H.264/MPEG-4 AVC is used. First, the human face regions in the video are detected and extracted. Then, ROIs are mapped into slice groups. After that, these regions are encrypted using selective video encryption based on chaos.

### **ROI coding with H.264/MPEG-4 SVC**

In addition to spatial, temporal and quality scalability the H.264/MPEG-4 SVC supports ROI scalability. This type of scalability is appropriate to many scalable video coding applications for instance, a mobile phone user may be required extracting only particular ROI in video, at same time other user with large portable device screen can extract other ROI to receive better video stream resolution. Thus, to fulfil these requirements, it would be necessary to transmit or store a scalable bit stream with different ROIs.

Grois *et al.* [167] present a scalable ROI video coding algorithm which enables the adaptation to the position, size and resolution of the ROI. This algorithm has two methods for ROI coding, the first is based on inter-layer prediction and the second the uses FMO. In the first proposed method, the authors cropped the ROI from the original sequence and used it as a base layer and increased the ROI resolution in the enhancement layer. After this, inter-layer prediction is applied in the cropping areas. Compared to conventional single layer coding, the method incurs in low bit rate overhead. However, this approach



does not allow the existence of non-ROI in the base layer and the size of ROI, in this layer, is constant along the sequence. In the second proposed method the ROI is encoded with FMO type 2 (only to enhancement layers) and each ROI is represented by a rectangular shape with the ROI and non-ROI regions coded in separate slices. However, this method only used rectangular ROI, and in most cases the ROI has not a rectangular shape. In addition, the algorithm does not implement the ROI detection method, here, the ROI is pre-defined by the user. Further, Lee *et al.* in [168] propose a scalable ROI algorithm (H.264/MPEG-4 SVC) which used the FMO with Box-Out method in the coding process. Two methods for ROI detection were proposed, the passive (pre-defined by the user) and active setting of ROI (based on motion vectors). The active selection method produced better subjective quality than the passive selection method. The algorithm supports Fine Grain Scalability (FGS) in ROI with low computing complexity in order to achieve better objective and subjective video quality.

### ROI coding for 3D video

The ROI coding is not restricted to the 2D video, the 3D video coding can also benefit from the use of ROI [163]. As mentioned before, ROI coding was originally developed for low bit rate video application since this coding mode is based on differentiated coding scheme that provide more bits and quality to the ROI and less quality and bits to the non-ROI. Due to the 3D video applications requirements and the ROI coding features, until now, few works have been published on ROI coding for 3D video.

Karlsson *et al.* [163] proposed a method to increase the perceived quality in a 3D video sequence at low bit rates. An ROI coding method was applied to 2D plus depth video sequence to guarantee a good quality in regions of the video that are more interesting to viewers. Here, the author defined the faces and the scene objects nearer the viewer as ROI. However, the nearer objects or the faces are not necessarily the most important regions of frame to viewers. Two ROI detection methods were proposed. In the first method the statistical properties of the depth information are examined. In the second method, the position of the ROIs in each frame is defined as the combination of the skin-colour detection (to detect the faces) map and depth detection map. A spatio-temporal filter was used to re-allocate bits from the non-ROI to the ROI. After this step, the filtered sequences were coded using the H.264/MPEG-4 AVC codec (Joint Model (JM) 10.1 high profile). Interesting results have been achieved by employing region-based techniques (detection and coding). However, a subjective assessment should be made to evaluate the

impact of this ROI coding approach in the overall quality of the video.

More recently, Pinto *et al.* [169] proposed the use of ROI as an extension of the concept of asymmetric coding for regions of different perceptual relevance in stereoscopic video. The results (objective and subjective assessment) show that it is possible to further exploit traditional asymmetric coding while maintaining the same perceptual quality in the stereoscopic video.

## 2.5.4 Performance metrics

In order to develop adequate flexible video coding, reliable quality assessment metrics are required for comparison purposes. After all, it is necessary to evaluate distortion introduced by coding, since all video coding standards, currently available, introduce distortion in coded video. For this, as explained back in Section 2.4.4, the performance metric can be classified in two types: subjective and objective, but only objective quality assessment will be considered here.

The most used and known objective metric to measure the distortion is the PSNR [170]. PSNR defined in Equation (2.25) is calculated on a logarithmic scale and depends on the Mean Squared Error (MSE) Equation (2.26) of between an original and an impaired image or video frame, relative to  $(2^n - 1)^2$  (the square of the highest-possible signal value in the image, where  $n$  is the number of bits per image sample). In the Equation (2.26),  $m$  and  $n$  are the image size and  $O$  and  $R$  are the original and reconstructed image respectively.

$$PSNR_{dB} = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (2.25)$$

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [O(i, j) - R(i, j)]^2 \quad (2.26)$$

PSNR can be calculated easily and quickly and is therefore a very popular quality measure, widely used to compare the “quality” of compressed and decompressed video images. MSE is by itself a distortion metric that quantifies the difference between two images or video frames. More recently, other objective metric to evaluate the distortion was proposed, it is based on the PSNR and was proposed Gisle Bjøntegaard proposed in [171] a model that measures the compression efficiency difference between two algorithms.

## 2.6 Discussion

In general the methods for flexible representation of visual information, still suffer from some limitations in the sense that they are not able to satisfy all challenges posed by 2D and 3D video applications. In order to overcome these limitations and issues that are still open for research the following topics are identified:

**Visual saliency computation methods** - although some significant work has been done in 2D video saliency, only few visual attention models have been proposed for 3D content and most of them are exclusively applied to image data [2, 37, 39, 40]. To the authors' knowledge, only two works can be used for 3D video [38, 41]. However, Zhang *et al.* [38] do not present a consistent objective validation of the results, which does not allow comparison with other methods. Iatsun *et al.* [41] proposed a comprehensive qualitative evaluation in this work, however Iatsun's method can be improved by adding face features. In order to address these issues a method is proposed, this was developed for 3D video but also applied for 2D video and 3D image which is explained in Chapter 3.

**SBD and key-frame extraction methods** - the selection of the features used by shot boundary and key-frame extraction methods is still an open research problem, because these features depend on the application, video content and representation format. For instance, in fast-motion scenes edge information is not the best choice to detect shot boundaries due to motion-induced blur. Thus, it may be better to automatically find the useful features based on some assumptions about the video-content. The majority of key-frame extraction methods published in the literature use low-level features and content sampling approaches to identify the relevant frames that should be included in the key-frame summary. Recently, the inclusion of perceptual metrics in the SBD and key-frame methods are gaining some space and in the context of 2D video, some key-frame extraction methods based on visual attention models have emerged as, [59, 62–64, 85]. However, for 3D video only one solution is available [111]. Hence, key-frame extraction in 3D video still poses relevant research problems to be investigated and efficiently solved.

Another open challenge is the combination of the visual features with additional information (audio features, text captions and content description) for use in the detection of shot boundaries and selection of the optimal frames in 3D video. Also, lacking are

summarisation methods (key-frame based or video skims), for some 3D video formats such as MVD and holoscopic video. Also open to research is the application of scalable summarisation to 3D formats [11]. In the context scalable summarisation several works were published which presented solutions for 2D video such as [172, 173].

**Evaluation of summarisation methods** - in the past evaluation frameworks for 2D key-frame summarisation methods were proposed in [174, 175]. More recently, Avila *et al.* [114] also proposed another evaluation setup, wherein the original video and the key-frame summaries of several methods are available for downloading, together with the results of several key-frame extraction methods for 2D video. Unfortunately for the case of 3D video, there is not as yet any similar framework, where key-frame summaries and the respective original sequences are available for research use. The number and diversity of evaluation metrics (objective, result description and subjective) used to compare state-of-the-art key-frame extraction methods make their comparative assessment a difficult task. Therefore, the development of metrics which can be used in the evaluation of key-frames summaries in different domains and 3D video formats is a very important area of video-summarisation related research. Furthermore the focus of the evaluation process must be application-dependent. For instance, in browsing applications, the time spent by the user to search or browse for a particular video is the most important factor, but on the other hand, in detection events, the evaluation metric must focus on the successful detection of these events. One other problem that arises in the evaluation process is the replication of results of previous works, as some works are not described with enough details to allow independent implementation or the input data is unavailable or else it is not easy to use due to data format incompatibilities or lack of information about their representation format. Thus, the best way to test and compare key-frame extraction methods for 2D and 3D is to build publicly accessible repositories containing test kits, made up of executable or web-executable versions of the methods and the test sequences.

**Key-frame presentation** - another challenging topic in the research of 3D key-frame summarisation is the design of an efficient and intuitive visualisation interface that allows easy navigation and visualisation of the key-frame summaries. These applications should be independent of the terminal capabilities (display dimension, processing and battery power), i.e., should be usable on small screen devices such as smartphones as well as UHD displays. In addition, the visualisation interface should be independent from the key-frame summarisation method, to allow the visualisation of different formats

of 3D key-frames video summaries, such as stereoscopic video or video-plus-depth and also 2D video in the same framework. The interface should be capable of dealing with the most common key-frame visualisation methods such as, static storyboard, dynamic slideshow and hierarchically arranged viewing. In particular, the most recent 3D interface for searching and viewing images or video in large databases, 3D-Ring and 3D-Globe, are interesting solutions which must be taken into account in the definition of new key-frame visualisation methods [7].

**Video summary coding** - in the past, the problem of scalable coding of video summaries was addressed in [176–178]. In [176] the authors propose a hierarchical frame selection scheme which considers semantic relevance in video sequences at different levels computed from compressed wavelet-based scalable video. In [177] a method to generate video summaries from scalable video streams based on motion information is presented, while in [178] the authors propose to partition a video summary into summarisation units related by the prediction structure and independently decodable. The existence of few studies and articles published about this subject in the literature, is one reason the development a method to code video summary. Thus, in the Chapter 6 it is proposed a method to encode an arbitrary video summary using dynamic GOP structures in scalable streams. The scalable stream obtained was fully compatible with the scalable extension of the H.264/AVC standard. However, all approaches were proposed for 2D video and used older generation video coding methods. The application of video summary coding to the 3D video format and the use of the most recent video coding, such as HEVC, should also be explored to find efficient coding tools for such purpose.

## 2.7 Conclusions

This chapter presents a review of several state-of-the-art methods for flexible representation and coding of visual information and some of the potential application of these methods and also the performance metrics used. Visual saliency, summarisation, retargeting and flexible video coding using scalability and ROI coding were studied in this context, as capable of providing some extra degree of flexibility beyond simple representation of pixel values. The relevance of visual saliency in summarisation, video retargeting and ROI coding was particularly emphasised. The critical review presented in this chapter also lead to identify some open research problems where innovative solutions may be further investigated.



# Visual saliency computation by feature aggregation

---

This chapter presents two methods for visual attention modelling which were developed with the goal to increase the flexibility of 3D visual information representation. Although originally developed for 3D content with some adaptations these methods can also be applied to 2D video, as shown in Chapter 4 and Chapter 5. The proposed methods are based on fusion of features maps which contain information from spatial, temporal (motion), depth dimensions and face detection. Several test videos annotated with eye-tracking data were used to validate the methods. The results demonstrate substantial performance gains in comparison to other state-of-the-art models.

These methods and part of the experiments presented in this chapter were published in J1, E1, C1, and C2.

## 3.1 Visual saliency computation methods

Recently, several methods to compute visual saliency were proposed for 2D video and 3D images, but for 3D video no definite solution exists. Therefore this is still an open research problem that requires approaches different from those used with 2D video or 3D images. The main novel aspects of the proposed methods are: (i) the flexibility of the method since it can be applied to different 3D content, such as stereoscopic images, video or video-plus-depth and 2D video; (ii) the comparisons with other methods based on publicly available fixation density reference datasets; (iii) better performance than other state-of-the-art models. The proposed methods are composed of four models used to compute features maps from information in different domains. In the first approach, the method combines the spatial, temporal and depth information in order to determine the visual saliency. The second approach improves the first method by adding the face

saliency map obtained by processing face features. The following sub-sections describe in details the proposed methods.

### 3.1.1 Visual saliency computation using spatio-temporal depth information

Figure 3.1 shows the functional diagram of the visual saliency computation method which uses spatio-temporal depth information. Either stereoscopic video or video-plus-depth can be used as 3D input formats with the former requiring a previous step of depth computation from the stereoscopic views. The method relies on computing three different maps, which representing relevant perceptual features from spatial (texture), motion and depth information. The spatial and motion feature maps,  $S_S$  and  $S_M$  respectively, are computed from a single view, while the depth feature map  $S_D$  is extracted from depth maps. Then, normalisation and fusion of these feature maps generates the intermediate map  $S'_G$ , which is subjected to a centre-bias weighting to generate the final 3D visual saliency map  $S_G$ .

#### Spatial saliency computation

Computation of the spatial feature map  $S_S$  is done based on a single view because this type of perceptual feature is mostly independent of the viewpoint for small baselines. Since spatial saliency computation has been thoroughly investigated by several authors in the past, the procedure used here has been inspired by several methods described in previous works [22, 29, 30]. These are chosen due to their performance and widespread use in recent research about visual attention models, thus being also useful as references for comparison.

Itti's model, proposed in [22], implements a hierarchical decomposition based on low-level features including intensity, colour and orientation. Thus, an input image is subsampled into a Gaussian pyramid and each pyramid level is decomposed into intensity, colour and orientation. Then each of the resulting maps are normalized and combined to form the final saliency map after ranking the focus of attention regions using a winner-takes-all approach. Instead of processing an image spatial domain, Hou's model [29] uses frequency domain. Thus, Hou and Zang proposed a spectral residual saliency model based on the idea that similarities imply redundancies. They propose that statistical



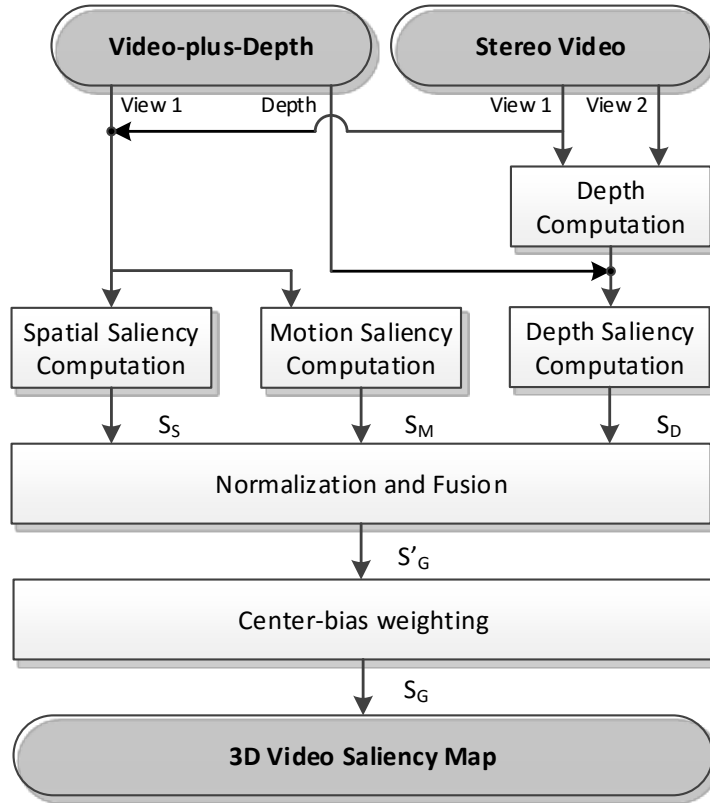


Figure 3.1: Functional diagram of the visual saliency computation method.

singularities in the spectrum may be responsible for anomalous regions in the image, where proto objects become conspicuous. Bruce’s AIM model, proposed in [30] is based on information maximisation and uses Shannon’s self-information measures to compute visual attention maps. This model is based on the premise that the visual saliency map is related to the amount of information provided, in regard to its local neighbourhood. Information of a visual feature  $X$  is define as  $I(X) = -\log p(X)$ , which is inversely proportional to the probability of observing  $X$  (i.e.,  $p(X)$ ). Thus, and according to Bruce’s AIM model, salient regions are the image areas where there are higher indecision and more self-information.

The proposed method allows the user to select which of these three visual attention models are used to compute the spatial feature maps  $S_S(i, j)$  with  $(i, j)$  being the pixel positions in a frame.

### Motion saliency computation

The motion feature map is computed based on the underlying idea that viewers tend to hold their gaze towards higher motion regions [32]. In this work, the motion feature map is computed from the motion intensity observed between two consecutive frames and computed using block matching over a search region. More precise motion estimation methods could be used to take into account the selective sensitivity of the HVS to motion and its complex and background motion accommodation capabilities [179], but these alternatives would result in higher computational complexities.

The motion intensity  $I(i, j)$  for each pixel  $(i, j)$  in a frame is computed as the magnitude of motion vectors:

$$I(i, j) = \sqrt{dx_{i,j}^2 + dy_{i,j}^2} \quad (3.1)$$

where  $dx_{i,j}$  and  $dy_{i,j}$  denote the two components of the motion vector along  $x$ -axis and  $y$ -axis, respectively. After computing the motion intensity for all pixels of each frame, the motion feature map is obtained by applying a  $5 \times 5$  spatial median filter to remove the estimation noise (i.e., outliers)

$$S_M(i, j) = \text{Median}(I(i, j), 5) \quad (3.2)$$

### Depth saliency computation

The depth feature map is computed based on the local depth contrast, following the reasoning that depth contrast is a dominant feature in depth perception [180]. The depth feature maps are computed as a sequence of two processing steps: *Depth contrast computation* and *Probability distribution modelling*.

*Depth Contrast Computation:* The depth contrast is obtained by filtering the depth map with a DOG filter which simulates the operation of the centre-surround mechanism of the HVS. The depth contrast  $C(i, j)$  at pixel position  $(i, j)$  is computed as,

$$C(i, j) = \text{DOG}(i, j) * D(i, j) \quad (3.3)$$

where  $D(i, j)$  is the depth map and  $DOG(i, j)$  is defined by

$$DOG(i, j) = \frac{1}{2\pi\sigma_1^2} \exp\left(-\frac{i^2+j^2}{2\sigma_1^2}\right) - \frac{1}{2\pi\sigma_2^2} \exp\left(-\frac{i^2+j^2}{2\sigma_2^2}\right) \quad (3.4)$$

The standard deviations  $\sigma_1$  and  $\sigma_2$  are filter parameters, which were set to 32 and 51 respectively, following the use of DOG for static 3D images described in [2].

*Probability Distribution Modelling* (PDM): This distribution quantifies the probability that a pixel at position  $(i, j)$  with depth contrast  $C(i, j)$  is gazed at. These probabilities are computed for all pixel locations of the depth contrast  $C(i, j)$ , by using the empirical data obtained from eye-tracking experiments carried out by Wang *et al.* and presented in [2], where, using the fixation information and the depth-contrast value for each pixel, a function is empirically derived from Bayesian principles, that quantifies the probability of the pixel being gazed at, when depth-contrast is  $x$ , i.e.,  $P(\text{pixel is gazed} \mid \text{depth contrast} = x)$ . Then, the depth feature map  $S_D(i, j)$  is given by a piecewise linear function that interpolates such empirical distribution for all  $C(i, j)$  within the perceptually relevant range.

### Normalisation and fusion

Before aggregation, the saliency feature maps computed in the previous steps are normalized to the range  $[0, 1]$  by division by the saliency peak value, which results in  $\hat{S}_S$ ,  $\hat{S}_M$  and  $\hat{S}_D$  obtained as follows.

$$\hat{S}_\alpha(i, j) = \frac{S_\alpha(i, j)}{\max_{\forall i, j}(S_\alpha(i, j))} \quad \alpha \in \{S, M, D\} \quad (3.5)$$

An aggregated visual saliency map  $S_{3D}$  is then computed as a weighted sum of the three normalized saliency feature maps, as given by Equation (3.6)

$$S'_G = w_s \hat{S}_S + w_m \hat{S}_M + w_d \hat{S}_D \quad (3.6)$$

where  $w_s + w_m + w_d = 1$ . These weights may be assigned the same default value, giving the same perceptual importance to all feature maps. However, following previous studies where motion was found as a predominant feature [181], a better choice may be to assign higher weight to the motion saliency map. Other fusion methods can be used, based on dynamic weights, saliency strength and energy [41], [182].

### Centre-bias weighting

Centre-bias weighting is used to model the human tendency to gaze at objects located in the centre of the visual scene. To model this behaviour the aggregated saliency map  $S'_G$  is further weighted with an image-centred two-dimensional Gaussian function  $W(i, j)$  that biases the visual saliency towards the image centre. Therefore, the final visual saliency map  $S_G(i, j)$  is defined for all pixels  $(i, j)$  of each frame by,

$$S_G(i, j) = S'_G(i, j)W(i, j) \quad (3.7)$$

where  $W(i, j)$  is given as,

$$W(i, j) = \exp\left(-\left\|\frac{L(i, j)}{\delta}\right\|^r\right) \quad (3.8)$$

and

$$L(i, j) = \sqrt{(i - M/2)^2 + (j - N/2)^2} \quad (3.9)$$

where  $L(i, j)$  is the Euclidean distance between pixel at spatial location  $(i, j)$  and the image centre and  $M$  and  $N$  are the video spatial dimensions. The  $r$  value used in Equation (3.8) is constant and  $\delta$  is given by:

$$\delta = c\sqrt{(M/2)^2 + (N/2)^2} \quad (3.10)$$

In this work, the values of  $r = 1.3$  and  $c = 1.7$  were used, following [183]. The saliency values of  $S'_G(i, j)$  vary in the range  $[0, 1]$ , where 1 and 0 indicate respectively the highest and the lowest possible saliency values.

### 3.1.2 Improved visual saliency computation method based on face saliency

The improved version of visual saliency detection method using spatio-temporal depth information is based on the computation and aggregation of four individual saliency feature maps, unlike the first approach, which only three individual saliency feature maps are used to compute final saliency map. The overall processing involved in the computation of the visual saliency estimate is illustrated in Figure 3.2. As the first approach,

this approach accepts 3D video in both stereoscopic and video-plus-depth format. The four saliency feature maps at the core of the proposed method are computed from the spatial (texture), temporal (motion), depth and face presence information and are then combined into a single saliency map. The spatial and motion saliencies, as well as the face saliency,  $S_S$ ,  $S_M$  and  $F_D$  respectively, are computed from the video texture, while the depth saliency map  $S_D$  is derived from the depth information. The pixel-wise values of these features are normalised and then aggregated resulting in a saliency value  $S'_G$ , which is further processed with a centre-bias weighting filter to compute the final saliency map  $S_G$ . This procedure is followed for each frame of the 3D video sequence thus producing a sequence of visual saliency maps.

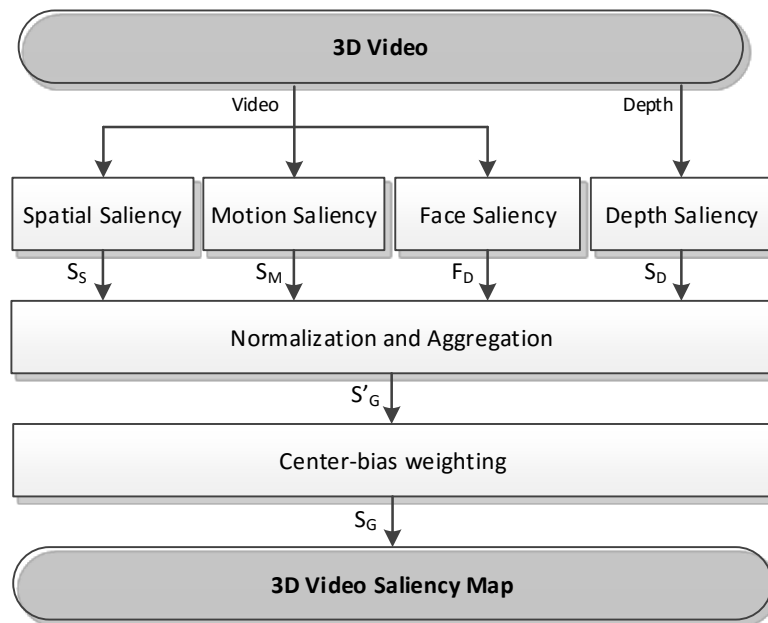


Figure 3.2: Functional diagram of the improved visual saliency computation method.

### Spatial, motion and depth saliency computation

The computation of the intermediate saliency maps  $S_S$ ,  $S_M$  and  $S_D$  are equal to the previous saliency computation approach, presented in the Section 3.1.1.

### Face saliency computation

A face detection algorithm is employed during the face saliency map  $F_D$  computation. This face detection algorithm produces a bitmap containing binary masks that indicate if

a given pixel is part of a face-region. Such mask is the face saliency map. Although other simpler methods, as face detection based on skin color could have been used, the Viola-Jones object detector algorithm [184] was chosen to detect and locate the human faces in each video frame. This object detector algorithm is based on the detection of specific features that contain information about the object to be detected such as faces, cars and others. This information can be encoded by Haar-like features which are sensitive to orientation of contrasts among image regions. For example, a human face can be represented as set features exhibiting the relationship of the contrast of different regions like eyes, nose, mouth etc. In order to explore the potentiality of Viola-Jones algorithm and to increase the number of face detected in each frame, a cascade object detector method is used. This method is based on cascade detection of the upper-body and face. Thus, a human face is considered detected if it is identified in the upper-body region. After face detection, the binary masks in the face saliency map (see Figure 3.3d) are used to identify regions of higher importance in the computation of the aggregated saliency map. Figure 3.3 shows an example of a face saliency map pertaining to a frame of the sequence *News report*.

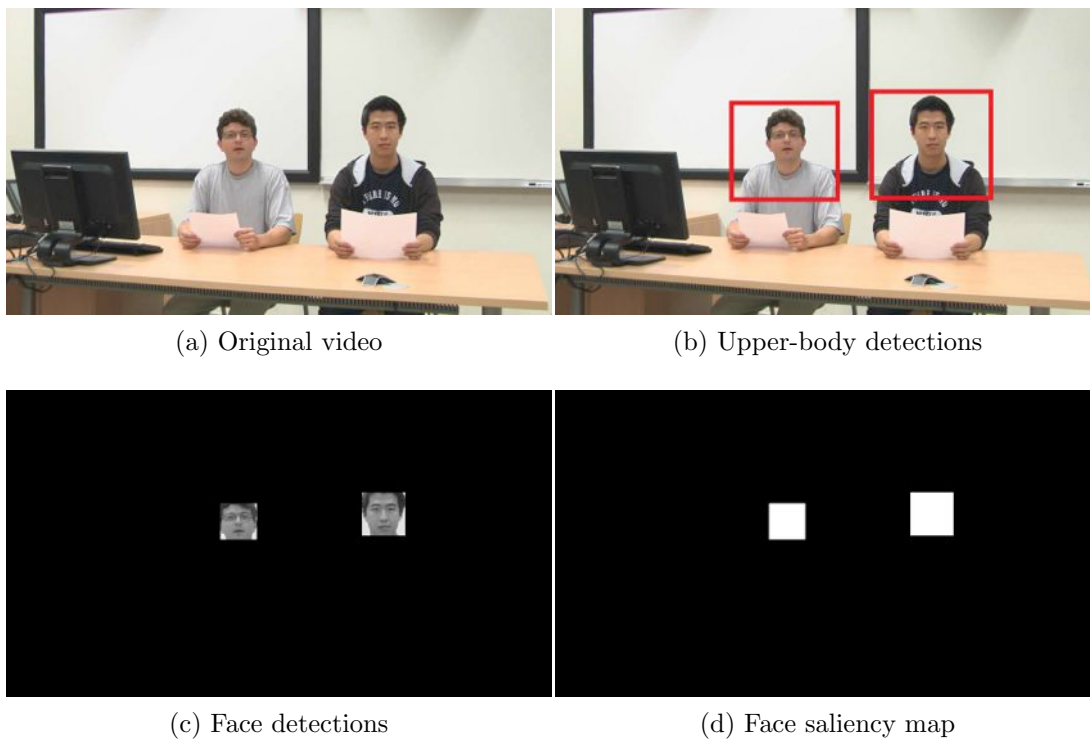


Figure 3.3: Example face saliency map of the *News report* sequence.

### Normalisation and aggregation

As in the previous approach presented in Section 3.1.1, before aggregation of the various saliency maps obtained in the previous steps, when necessary these are normalized to the range  $[0, 1]$  yielding  $\hat{S}_S$ ,  $\hat{S}_M$  and  $\hat{S}_D$  as follows.

$$\hat{S}_\alpha = \frac{S_\alpha(i, j)}{\max_{\forall i, j}(S_\alpha(i, j))} \quad \alpha \in \{S, M, D\} \quad (3.11)$$

The aggregated visual saliency map  $S'_G$  is then computed as a weighted sum of the three saliency maps and mask  $F_D$ , as given by equations (3.12).

$$S'_G = \sum_{\alpha \in \{S, M, D\}} w_\alpha \hat{S}_\alpha + w_f F_D, \quad \text{with} \quad \sum_{\alpha \in \{S, M, D\}} w_\alpha + w_f = 1 \quad (3.12)$$

### Centre-bias weighting

Similarly to the computation of the intermediate saliency maps  $S_S$ ,  $S_M$  and  $S_D$ , the application of the Centre-bias weighting in the process is similar to the previous saliency computation approach, presented in the Section 3.1.1.

## 3.2 Results and analysis

To evaluate the performance of the proposed methods several experiments were realised, where the algorithms described so far were used to compute the visual saliency of several 3D test videos. In the next sections, the methodology used to evaluate the proposed methods is described and results are also analysed and discussed.






### 3.2.1 Experimental setup and methodology

The experiments are divided into two parts. In the first part, the first approach of proposed method is evaluated by comparing the computed saliency maps against six other competing methods for the computation of 3D visual saliency maps. The performance of all methods was quantified using as ground-truth for the fixation density provided by the maps from [1]. The influence of centre-bias weighting is also studied and a visual

comparison between the saliency and FDM is as well made. In second set of experiments, the second method was evaluated by comparing the computed saliency map for 3D video with a ground-truth set of fixation density maps created from eye-tracking experiments, publicly available [1] [3]. The influence of face saliency map is investigated and a visual comparison between the saliency and FDM is also made.

The performance evaluation was measured by the PLCC and KLD values of the computed saliency and FDM values series. Five different 3D video sequences were chosen to evaluate the two proposed methods: *Boxers*, *Hall*, *Phone call*, *Laboratory* and *News report*. Table 3.1 lists the tests video sequences and their characteristics. Figure 3.4 shows the first frame of the original video and Fang’s [3] and Hanhart’s [1] FDM data respectively, for the *Boxers*, *Hall*, *Phone call*, *Laboratory* and *News report* sequences.

Table 3.1: Details of the test sequences.

	Name	#Frames	Resolution	#Frames containing faces
	Boxers	250	1920x1080@25fps	20
	Hall	250	1920x1080@25fps	0
	Phone call	250	1920x1080@25fps	140
	Laboratory	250	1920x1080@25fps	70
	News report	250	1920x1080@25fps	250

In first set of experiments, the proposed method was used visual saliency maps are based on fusion of the three saliency feature maps followed by centre-bias weighting. The weights  $w_s$ ,  $w_m$  and  $w_d$  of Equation (3.6), were assigned constant values, following the underlying idea that motion features are more relevant than others [181]. It was found empirically that the results are not critically dependent on small variations of these weights and after experimentation the values  $w_s = w_d = 0.25$ ,  $w_m = 0.5$  were chosen to be used in the remaining experiments. The motion field required to compute the saliency map  $S_M$  was estimated using block-match methods with a block size equal to  $[16 \times 16]$ .

In second set of experiments, the weights used in Equation (3.12),  $w_S$ ,  $w_M$ ,  $w_f$  and  $w_D$  were the same for all frames. Experimentation showed that small variations in these weights do not alter drastically the results and that values  $w_S = w_M = w_D = w_f = 1/4$



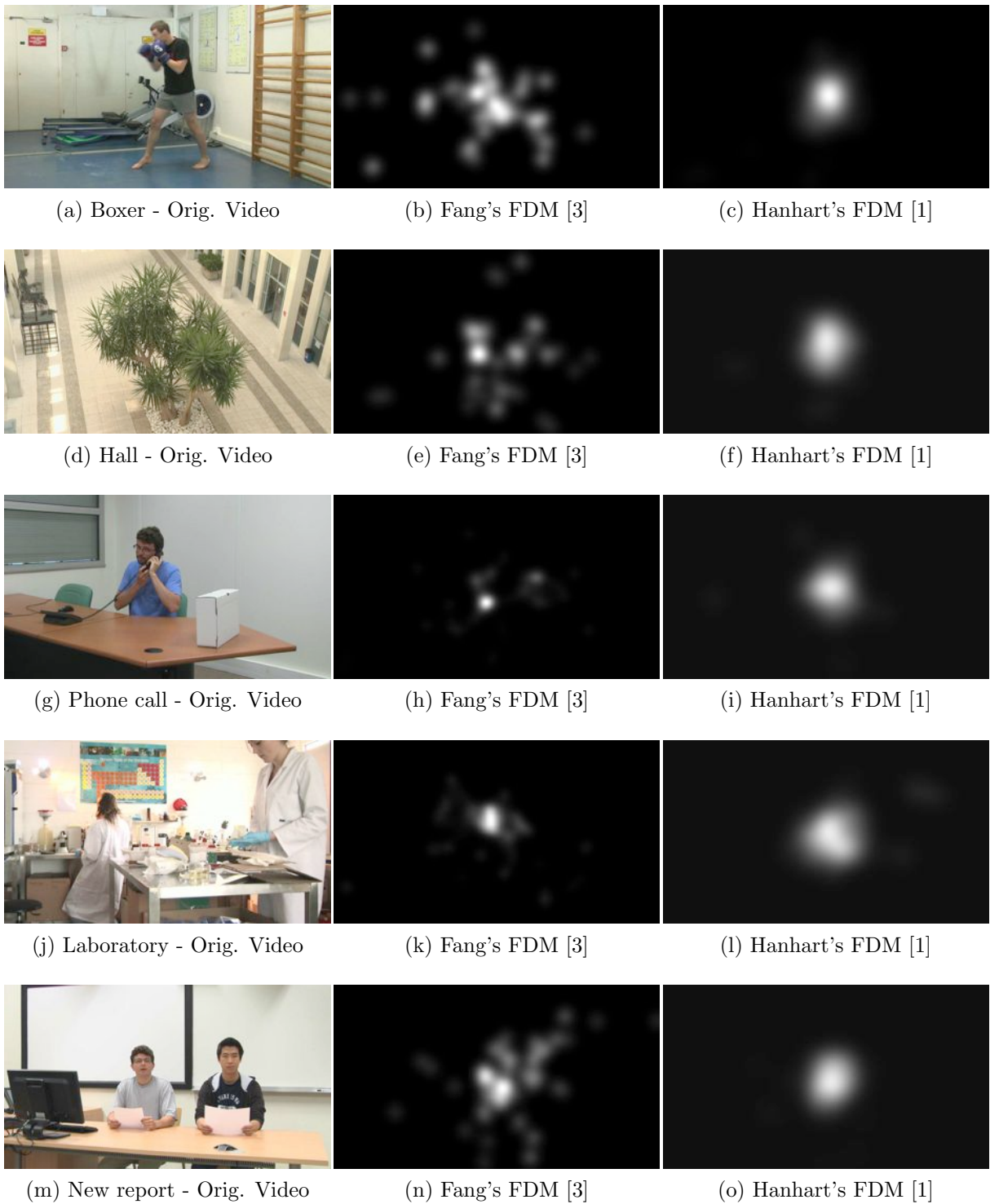


Figure 3.4: First frame of the original video, Fang's [3] and Hanhart's [1] FDM data respectively.

were a good choice. When face detection is not used the weight  $w_f$  is 0 and the remaining weights are equal to  $w_S = w_M = w_D = 1/3$ .

### 3.2.2 Visual saliency computation using spatio-temporal depth information

Tables 3.2 and 3.3 present the results of the PLCC and KLD evaluation obtained for five 3D video sequences using nine different methods to compute 3D visual saliency maps, including the proposed one. The three state-of-the-art methods from Itti [22], Hou [29] and Bruce [30] are used as references for comparison. These are 2D image-based methods, which were also used by Wang [2] to compute spatial saliency maps for 3D images. The results of our proposed method (Prop.) are also presented in Tables 3.2 and 3.3 separated into three different cases, according to the method used to compute the spatial saliency maps  $S_S$ , Itti, Hou or Bruce's, as described Section 2.2.1. The image-based methods were applied to all frames of the test video sequences.

The results in Tables 3.2 and 3.3 evidence a consistent relationship between the PLCC and KLD values for these sequences, since low values of PLCC normally correspond to high values of KLD and vice-versa. One can notice that the performance of the various methods increase with the number of features included in the models. For instance, visual saliency maps obtained using Wang's method [2] are better correlated with the FDM than simple 2D texture modes. This is mostly due to the inclusion of depth related features in the computation of the 3D images saliency maps. It is noteworthy that the proposed method produces better results than Wang's. Probably this increase in performance is due to the higher number of features used in the method proposed here. The inclusion of the motion feature map and centre-bias weighting function result in a better model of the human visual perception process and leads to a higher similarity between the computed saliency maps and ground truth FDM of 3D video test sequences.

The average values of PLCC and KLD shown in Tables 3.2 and 3.3 show that for the three cases under study the proposed method achieves better results than the competing methods. Overall, the maximum average PLCC (0.488) and the minimum average KLD (0.579) are achieved for the proposed method using Hou's spatial feature maps, i.e., *Prop.(Hou)*. This method has similar PLCC and KLD values for *Boxers*, *Hall* and *Phone call* test sequences. In comparison to *Prop.(Bruce)* the *Prop.(Hou)* method has better values of PLCC and KLD. For sequence *Laboratory* the PLCC achieved by *Prop.(Bruce)*

Table 3.2: PLCC evaluation-proposed and competing methods - Hanhart’s FDM data [1].

Method	Itti [22]	Hou [29]	Bruce [30]	Wang (Itti)[2]	Wang (Hou)[2]	Wang (Bruce)[2]	Prop. (Itti)	Prop. (Hou)	Prop. (Bruce)
Boxers	0.185	0.247	0.282	0.266	0.307	0.315	0.602	0.654	0.582
Hall	0.123	0.357	0.237	0.197	0.361	0.239	0.333	0.451	0.381
Phone call	0.297	0.526	0.386	0.307	0.527	0.431	0.547	0.584	0.516
Laboratory	0.09	0.081	0.324	0.166	0.118	0.326	0.264	0.262	0.385
News report	0.413	0.449	0.424	0.414	0.457	0.401	0.404	0.492	0.403
Average	0.222	0.332	0.330	0.270	0.354	0.342	0.430	<b>0.488</b>	0.453

Table 3.3: KLD evaluation-proposed and competing methods - Hanhart’s FDM data [1].

Method	Itti [22]	Hou [29]	Bruce [30]	Wang (Itti)[2]	Wang (Hou)[2]	Wang (Bruce)[2]	Prop. (Itti)	Prop. (Hou)	Prop. (Bruce)
Boxers	1.449	0.566	2.547	1.177	0.607	2.031	0.828	0.713	1.327
Hall	1.732	0.639	3.427	1.435	0.638	2.626	1.257	0.596	1.687
Phone call	1.207	0.704	2.039	1.133	1.097	1.419	0.502	0.409	0.798
Laboratory	1.205	0.766	2.934	1.445	0.700	2.107	1.188	0.802	1.497
News report	0.876	0.391	2.925	1.051	0.399	2.518	1.040	0.375	1.975
Average	1.294	0.613	2.774	1.248	0.688	2.140	0.963	<b>0.579</b>	1.457

is the best (i.e., 0.385), while for KLD, the method Wang(Hou) yields the best result (i.e., 0.700)

These results show that the proposed method tends to show better PLCC and KLD figures, most likely due to the use of more visual feature maps and of the centre-bias weighting in the computation of the 3D video visual saliency map. This means, that the visual saliency maps computed by the proposed method tend to better match the FDM obtained from human viewers. Since the PLCC and KLD values obtained from all methods are still far from their theoretical maxima there is still significant margin for improvement.

### Visual comparison between saliency maps and FDM

Figure 3.5 shows the visual results of the proposed method and *Phone call* sequence. For each figure, single frames of the original video, depth map, Hanhart’s FDM [1] and different saliency maps are presented where the magnitude of the visual saliency is coded with white indicating high saliency and black representing very low saliency. It can be observed that in Figure 3.5 the *Prop.(Hou)* method provides results that are significantly closer to the FDM data, i.e., the ground-truth (Figure 3.5c).

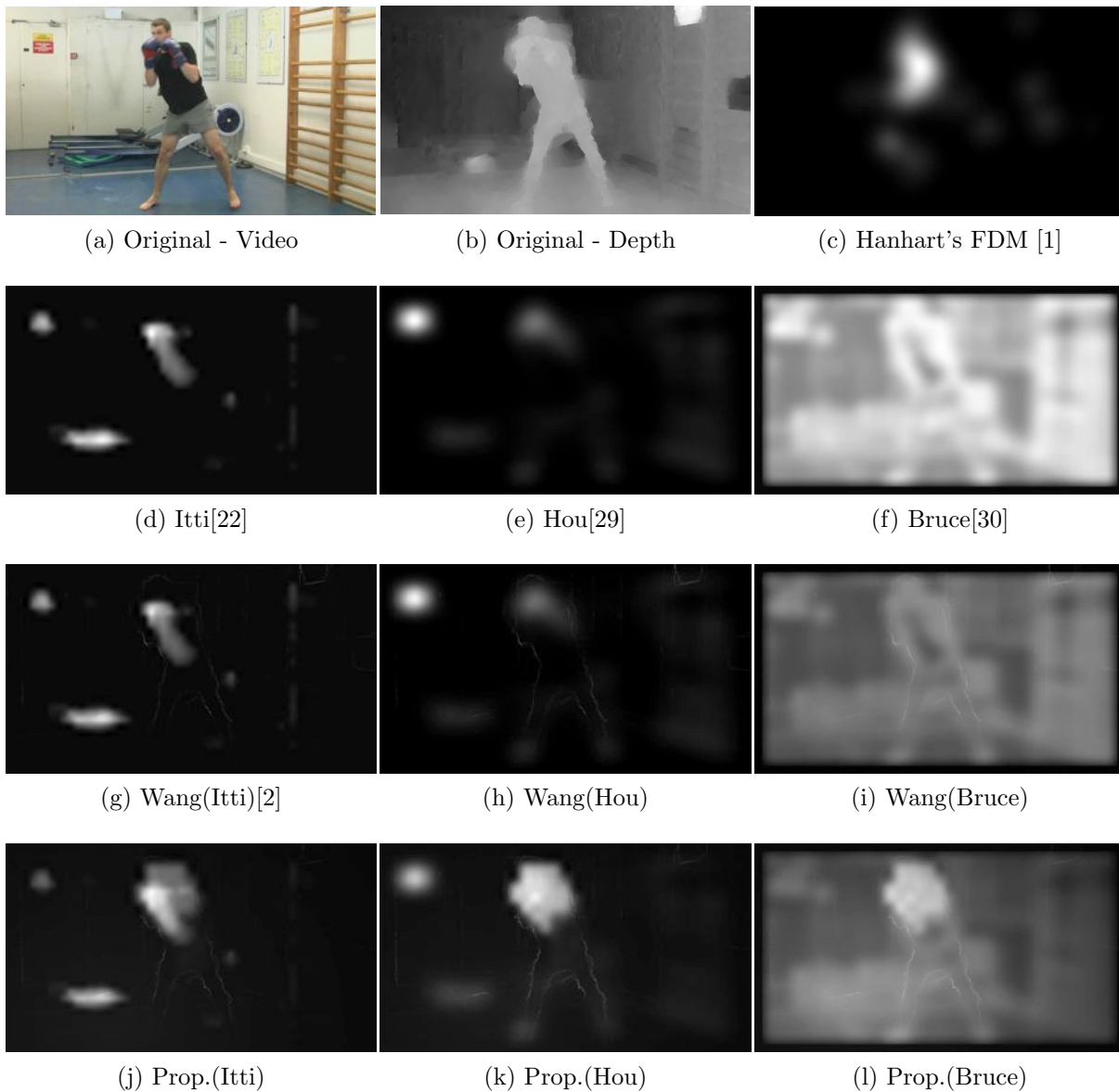


Figure 3.5: Visual saliency - Prop. and competing methods for the frame 120 of the *Boxers* sequence.

### The influence of centre-bias weighting

To better understand the role of centre-bias weighting in the results obtained by the proposed method, further tests were run specifically for this purpose. In these tests only the spatial saliency maps were used, which is equivalent to defining the weight values  $w_i$  of the Equation (3.6) as  $w_s = 1$  and  $w_m = w_d = 0$ . Table 3.4, shows the results obtained from the proposed method using the centre-bias weighting function with the

three methods from Itti [22], Hou [29] and Bruce [30]. The reference for comparison are the same methods without the centre-bias weighting function, the results of this methods are presented in the Table 3.3 and 3.2 for the PLCC and KLD evaluation respectively.

Table 3.4: Centre-bias weighting performance with Hanhart’s FDM data [1].

Method Metric	Prop.(Itti)		Prop.(Hou)		Prop.(Bruce)	
	PLCC	KLD	PLCC	KLD	PLCC	KLD
Boxers	0.192	1.418	0.274	0.565	0.308	2.015
Hall	0.226	1.672	0.407	0.553	0.364	2.418
Phone call	0.308	0.805	<b>0.540</b>	<b>0.11</b>	0.331	1.429
Laboratory	0.097	1.625	0.093	0.373	0.354	2.105
News report	0.411	0.871	0.453	0.379	0.432	2.802
Average	0.247	1.278	0.353	0.396	0.358	2.154

These results reveal that inclusion of the centre-bias weighting function in the proposed method increases its performance, as shown by the better PLCC and KLD values observed when centre-bias weighting is used. From the results in Table 3.4 one can find that the average gain of PLCC lies between 6% to 11% while that of KLD lies between 1% to 35%. The best performances are obtained from *Prop.(Hou)* method with *Phone call* sequence. Overall, the PLCC and KLD gains show that it is beneficial to include centre-bias weighting function in the proposed method.

### 3.2.3 Improved visual saliency computation method based on face saliency

Tables 3.5 and 3.6 show the results obtained for six different saliency computational methods, with two FDM databases, Hanhart’s [1] and Fang’s [3] respectively. The results of Wang’s method presented in the first six columns of the Tables 3.5 and 3.6 are used as reference for comparison with the proposed method without face detection (Tables 3.7 and 3.8) and with face detection (second six columns of Tables 3.5 and 3.6). In these two tables, the first, second, seventh and eighth column list the results obtained from frame-level spatial saliency maps  $S_S$  using Itti’s method [22], the third, fourth, ninth and tenth column using Hou’s method [29] and the fifth, sixth, eleventh and twelfth using Bruce’s [30].

The average values of PLCC and KLD presented in the Tables 3.5 and 3.6 show that proposed method with face saliency, for the three spatial saliency maps, achieves better results than the Wang’s method for the two FDM databases. However, the average values

Table 3.5: Performance of the proposed method with face saliency *vs* Wang’s method [2]-Hanhart’s FDM data [1].

Method Metric	Wang(Itti)		Wang(Hou)		Wang(Bruce)		Prop.(Itti)		Prop.(Hou)		Prop.(Bruce)	
	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD
Boxers	0.197	0.875	0.255	0.856	0.285	2.163	0.531	0.863	0.605	0.797	0.572	1.515
Hall	0.129	1.909	0.357	0.886	0.238	2.791	0.261	1.378	0.418	0.629	0.375	1.983
Phone call	0.210	1.460	0.549	1.056	0.24	1.495	0.252	0.565	0.342	0.583	0.264	0.945
Laboratory	0.010	1.484	0.080	1.540	0.326	2.264	0.035	1.288	0.090	0.985	0.302	1.606
News report	0.414	1.251	0.309	1.399	0.424	2.518	0.431	1.466	0.498	0.972	0.479	1.854
Average	0.192	1.396	0.310	1.147	0.303	2.246	0.302	1.112	0.391	0.793	0.398	1.581

Table 3.6: Performance of the proposed method with face saliency *vs* Wang’s method [2]-Fang’s FDM data [3].

Method Metric	Wang(Itti)		Wang(Hou)		Wang(Bruce)		Prop.(Itti)		Prop.(Hou)		Prop.(Bruce)	
	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD
Boxers	0.004	0.560	0.032	0.364	0.047	3.092	0.041	0.558	0.020	0.488	0.125	2.196
Hall	0.013	1.096	0.115	0.455	0.100	3.644	0.024	0.462	0.103	0.429	0.069	2.736
Phone call	0.034	1.470	0.163	0.926	0.131	2.621	0.155	0.513	0.209	0.524	0.161	1.883
Laboratory	0.018	1.432	0.021	0.772	0.104	3.480	0.066	0.609	0.053	0.467	0.168	2.630
News report	0.001	4.369	0.001	4.552	0.005	1.244	0.010	3.570	0.015	3.090	0.003	0.628
Average	0.014	1.785	0.066	1.414	0.077	2.816	0.059	1.142	0.080	1.000	0.105	2.015

of proposed method with face saliency obtained on Hanhart’s FDM database [1] are better than the PLCC and KLD values obtain by the Fang’s FDM database [3].

The improved performance achieved by the proposed method with face saliency is evidenced by the higher PLCC and lower KLD values for the two FDM databases as it is shown in Tables 3.5 and 3.6, when compared with the proposed method without face saliency map, listed in Tables 3.5 and 3.7 for Hanhart’s FDM database and Tables 3.6 and 3.8 for Fang’s FDM database. Thus, from the results in Tables 3.5 and 3.7 one can find that the average gain of PLCC lies between 11% to 16% while that of KLD lies between 1% to 4% for Hanhart’s FDM database. For Fang’s FDM database, the average gain of PLCC ranges from 5% to 20% while that of KLD ranges from 1% to 3%.

For sequences with a high number of human faces (e.g., *News report*, the proposed method with face saliency achieves better results than the other two approaches. For sequences with low number of faces (e.g., *Hall*, *Boxers*) the results are quite similar in both cases, regardless the use of face detection, as expected. Face definitely attracts human visual attention and the results in Tables 3.5 and 3.7 (Hanhart’s FDM database) and Tables 3.6 and 3.8 (Fang’s FDM database) provide evidence of that fact.

Table 3.7: Proposed method without face saliency map-Hanhart’s FDM data [1].

Method	Prop.(Itti)		Prop.(Hou)		Prop.(Bruce)	
Metric	PLCC	KLD	PLCC	KLD	PLCC	KLD
Boxers	0.533	0.865	0.604	0.789	0.501	1.535
Hall	0.261	1.378	0.418	0.629	0.375	1.983
Phone call	0.272	0.577	0.325	0.621	0.259	0.955
Laboratory	0.035	1.286	0.091	0.985	0.302	1.612
News report	0.202	1.489	0.316	1.041	0.347	2.150
Average	0.261	1.119	0.351	0.813	0.357	1.647

Table 3.8: Proposed method without face saliency map-Fang’s FDM data [3].

Method	Prop.(Itti)		Prop.(Hou)		Prop.(Bruce)	
Metric	PLCC	KLD	PLCC	KLD	PLCC	KLD
Boxers	0.039	0.561	0.019	0.497	0.123	2.232
Hall	0.024	0.462	0.103	0.429	0.069	2.736
Phone call	0.107	0.536	0.171	0.496	0.140	2.024
Laboratory	0.064	0.610	0.052	0.470	0.167	2.644
News report	0.004	3.621	0.002	3.151	0.003	0.718
Average	0.048	1.158	0.069	1.009	0.100	2.071

### Visual comparison between saliency maps and FDM

Figure 3.6 shows the visual results of our proposed method without and with face saliency map, for the *Boxers* sequence. It can be observed that in Figure 3.6 the *Prop.(Hou)* method provides results that are significantly closer to the FDM obtained from the eye-tracking experiments, i.e., the ground-truth (Figures 3.6c and 3.6b).

## 3.3 Conclusions

This chapter presents two methods to compute visual saliency maps for 3D video based on fusion of saliency feature maps followed by a centre-bias weighting function. These computational methods presented and analysed in the previous sections uses the two 3D video FDM publicly available [1, 3] to validate their results. The experimental results show that the proposed methods outperform methods introduced by other authors. Due to their modular features that allow inclusion or not of specific saliency feature maps, our methods can be applied to different types of 3D content format, such as stereoscopic images and video, as well as video-plus-depth. In this work, these two methods will be used, with some adaptation, in video summarisation and UHD retargeting methods described in Chapters 4 and 5. In addition to these applications the methods are also suitable for perceptual 3D video coding, quality evaluation and robust coding using regions of interest, among others.

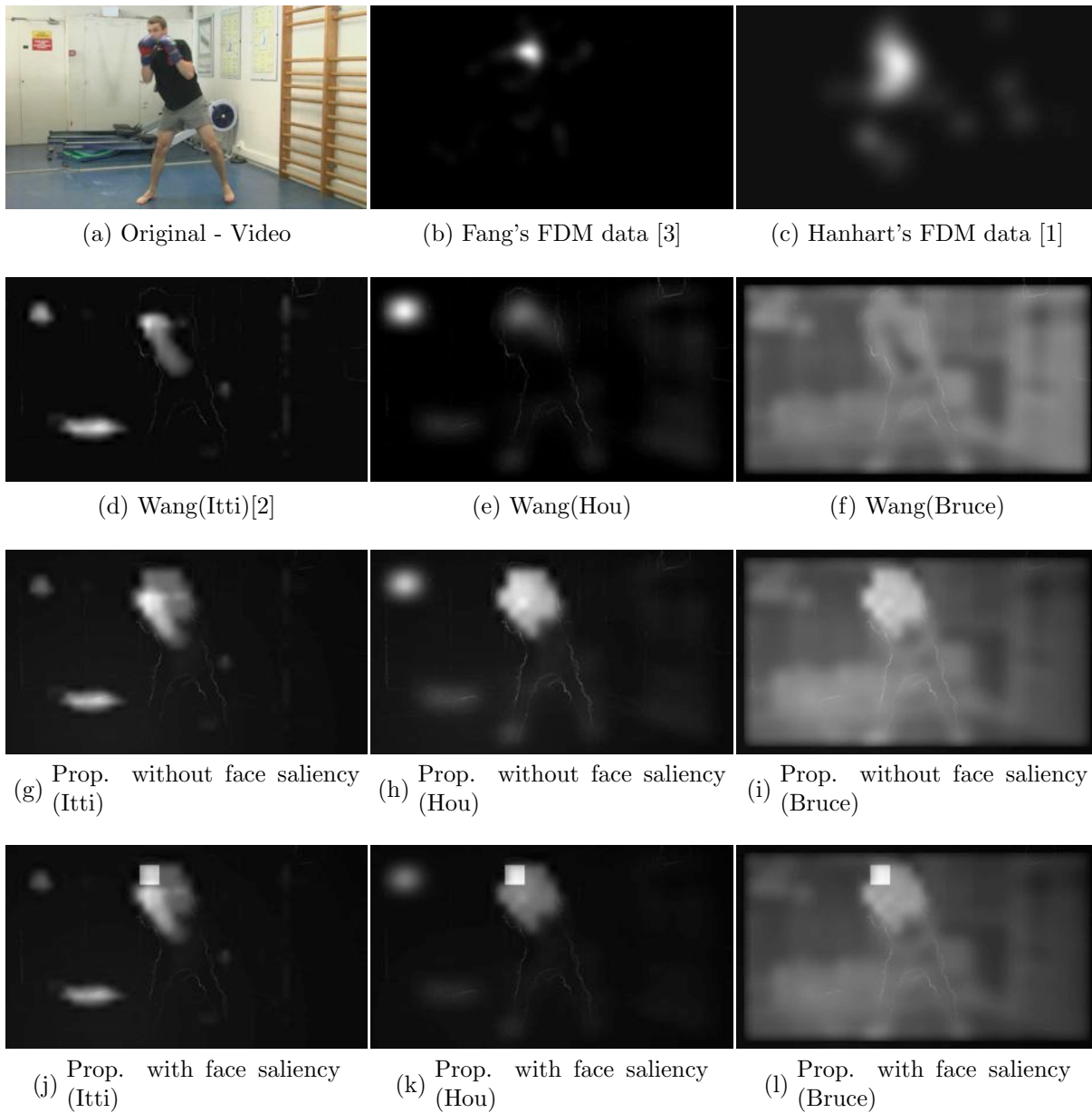


Figure 3.6: Visual saliency - proposed method with and without face saliency maps *vs* competing methods for the frame 120.



# Video retargeting

---

This chapter presents a study on spatio-temporal video retargeting methods using visual saliency to find a cropping window with the most important visual information contained in high resolution images. For instance, resizing UHD video down to smaller resolutions, such as those supported by mobile devices, can benefit from this type of retargeting. The proposed methods are capable of dealing with different input video resolutions in order to produce adaptive representations of the visual content, according to its importance as predicted by saliency models. Furthermore, the proposed retargeting methods limit the jitter between consecutive frames by applying time-domain filtering to guarantee that the most important content is preserved while the cropped spatial window location and size are stable.

Visual comparison between the results obtained from the proposed retargeting methods and other non-content-aware methods, including seam carving, is presented for relative performance evaluation. Additionally, the dynamics of the temporal evolution of retargeted frames in regard to their relative position in the higher definition frames, is also utilized to evaluate the performance of the jitter attenuation filter. The influence of temporal consistency on coding efficiency is further studied in this chapter and the results of these evaluations show that the proposed methods achieve good performance and have potential application in seamless access to UHD content by any type of device with reduced screen resolution. These methods and part of the experiments presented in this chapter were published in E1.

## 4.1 Spatio-temporal adaptation method based on visual saliency information

The video retargeting method presented in this section consists of a spatio-temporal adaptation and is broken down into four main steps, as shown in the Figure 4.1: (i) saliency

map computation based on attention modeling; (ii) identification of the cropping window which encloses the most salient part of the frame; (iii) improvement of temporal consistency between consecutive frames through temporal filtering of the spatial location of the cropping window; (iv) cropping of the original sequence based on the location and size of the cropping window to obtain the retargeted video. The proposed method resizes a video from resolution  $W \times H$  to  $W' \times H'$ , where  $W$  and  $H$  are the width and height of the original video, and  $W'$  and  $H'$  are the width and height of the retargeted video.

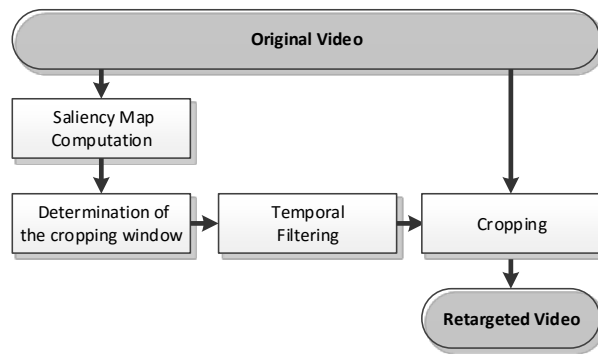


Figure 4.1: Functional diagram of the video retargeting method.

### 4.1.1 Visual saliency map computation

The starting point of the proposed retargeting procedure is the computation of a saliency map based on the method presented in Section 3.1.2, which aggregates three saliency feature maps, computed from the spatial (texture), temporal (motion) and face presence information. Figure 4.2 shows the functional diagram of the saliency computation method video, where the computation of the saliency features maps  $S_S$ ,  $S_M$ ,  $F_D$  as well as the normalization and aggregation and center-bias weighting function are explained in Section 3.1.2. Hou’s model [29] was used to process saliency feature maps  $S_S$ . The motion field required to compute the saliency map  $S_M$  was estimated using block-matching methods with a block size equal to  $[16 \times 16]$ . The face saliency map  $F_D$  is based on the Viola-Jones algorithm [184], which is used to detect and locate the human faces in each video frame. These saliency feature maps are combined into a single saliency map (one per frame) with the same resolution as the original video sequence. For example, Figure 4.3b shows the saliency map obtained for the *Jockey* sequence, where the white regions are the most relevant and the white square represents the region of the jockey face.

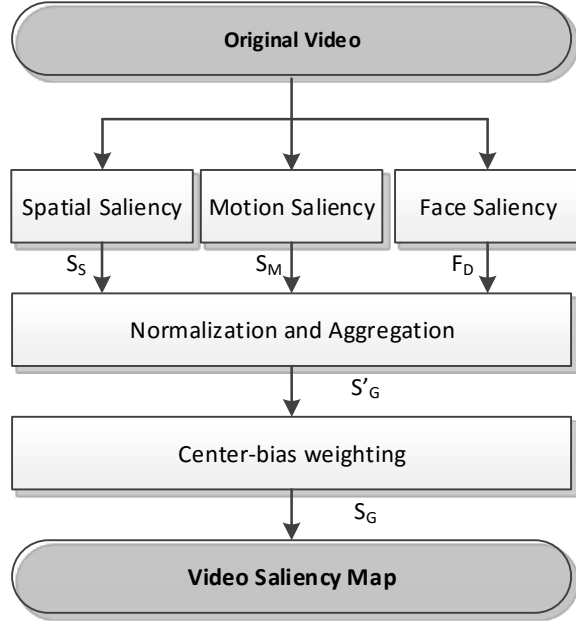


Figure 4.2: Functional diagram of visual saliency computation method.

#### 4.1.2 Determination of the cropping window

The cropping window of frame  $n$  is defined as  $c_n(x, y, Cw, Ch)$ , where  $x$  and  $y$  are the spatial coordinates of the upper-left corner of the cropping window in the original frame, and  $Cw$  and  $Ch$  are the width and height of the cropping window. The goal of this step is to identify the cropping window  $c_n$  in the saliency map  $S_{G_n}$  with highest energy. The (saliency) energy of the cropping window  $E(c_n)$  of the frame  $n$  is defined as:

$$E[c_n(x, y, Cw, Ch)] = \sum_{i=x}^{x+Cw-1} \sum_{j=y}^{y+Ch-1} S_{G_n}(i, j)^2 \quad (4.1)$$

$$0 \leq x \leq W - (Cw - 1), 0 \leq y \leq H - (Ch - 1)$$

where  $S_{G_n}(i, j)$  is the saliency map value for position  $(i, j)$  of frame  $n$  and  $W$  and  $H$  are the width and height of the original video sequence. The search for the cropping window of the saliency map  $S_G$  with highest energy is based on an exhaustive search process. For each frame, a unique cropping window with fixed size (see red box in the Figure 4.3) is found which will define which region of original sequence will be retained as a retargeted frame.

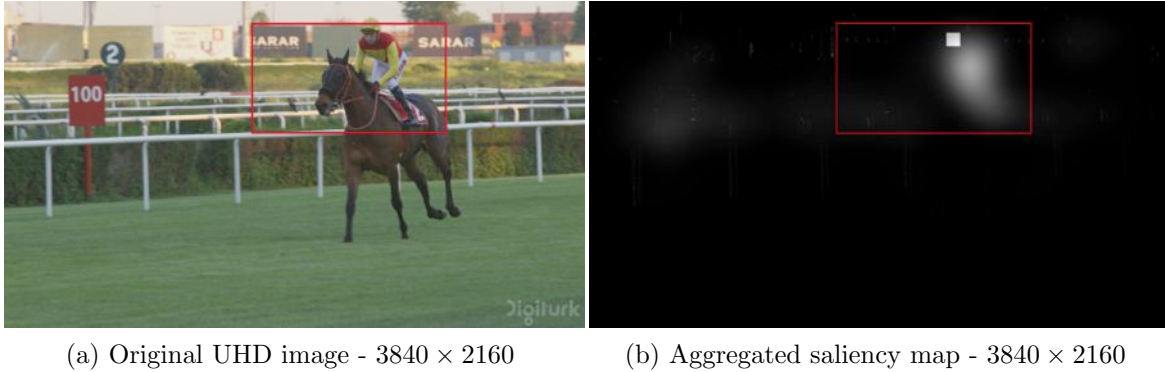


Figure 4.3: Red box is the cropping window ( $1280 \times 720$ ) of the *Jockey* sequence.

### 4.1.3 Temporal filtering

In video retargeting methods the preservation of the visual important window of the original video is necessary but not sufficient to obtain a good quality retargeted video. Another important requirement is that the content of cropping window does not change a lot between frames, i.e., that the cropping window displacements over the original video frame space should be as smooth as possible without sudden changes. One possible way to limit the magnitude of the changes in the spatial location of the cropping window across successive frames is by application of a temporal filtering step to the location of the cropping window, hopefully improving the stability of the retargeted video. To achieve this goal a median filter is used to filter the consecutive values of the cropping window coordinates. The option for a median filter is justified by the following facts: firstly, the median of a set of  $n$  values is always one value from the set, something that does not always occur with an average filter and secondly, the median filter is less sensitive to errors or to the extreme values than the average filter [185].

Consider an original video sequence with  $n$  frames and a set of cropping window parameters  $C$ , denoted by  $C = \{c_0, c_1, \dots, c_{n-1}\}$ , where the subscripts represent the temporal order of frames and the elements are vectors defining the position and size of the cropping windows. Let  $C_x$  and  $C_y$  be two vectors made of the  $x$  and  $y$  coordinates of the upper-left corner of the  $n$  cropping windows. The final spatial locations of the cropping windows are obtained by applying a median filter of size 15 to the  $C_x$  and  $C_y$ , as per Equation 4.2 where  $\hat{C}_x$  and  $\hat{C}_y$  are filtered version of  $C_x$  and  $C_y$ . Note that, in this case, the median filter is not applied to the  $Cw$  and  $Ch$  cropping window parameters as the sizes of these windows are constant and have the same value for all frames of the video sequence.

$$\begin{aligned}\hat{C}_x &= \text{Median}((C_x), 15) \\ \hat{C}_y &= \text{Median}((C_y), 15)\end{aligned}\tag{4.2}$$

Figure 4.4 shows the temporal evolution of the coordinates of the upper-left corner of the cropping window (horizontal and vertical position). The blue and red lines represent the  $x$  and  $y$  spatial positions before and after the median filtering of size 15. As shown in Figure 4.4, after jitter removal, the displacements of the cropping window follow a much smoother trajectory when compared to that of the unfiltered case.

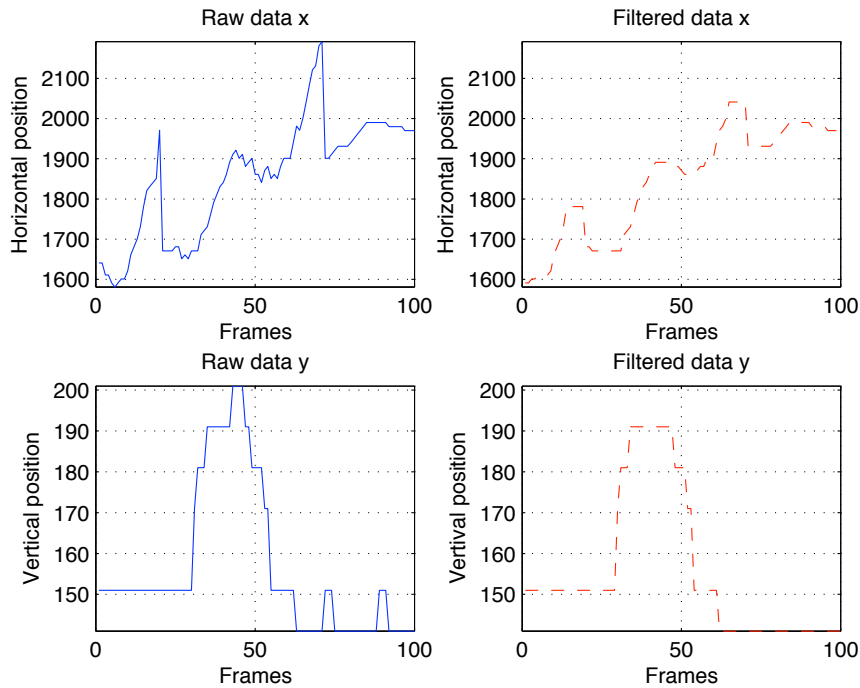


Figure 4.4: Temporal evolution of the upper-left corner of the cropping window - *Jockey* video sequence.

#### 4.1.4 Cropping

The last step of the method proposed is the cropping of the original video according to the cropping windows location and size parameters where the location information has been filtered as described in the previous section. Figure 4.8f shows one example of cropped frame for *Jockey* video sequence.

## 4.2 Hybrid video retargeting method based on visual saliency information

In the method proposed in the previous section the retargeting approach was based on the use of a maximum saliency energy criterion to define the position of the cropping window on the original frame. However this approach has some problems as in some cases important objects with high saliency values are not included in the final retargeted frame. For example, in Figure 4.9f the boat is not totally included in the retargeted frame, the same thing happening in Figure 4.8f where the head of the jockey is partially cut. In order to solve this problem, an hybrid retargeting method is proposed which involves a cropping step with a variable sized window followed by a resolution changing operation to fit the cropped window to the desired resolution. Figure 4.5 shows the main steps of this improved video retargeting method, in which some operations, like the visual saliency map computation, are the same as those used in the retargeting method of the previous section. The other operations are explained in the next sub-sections.

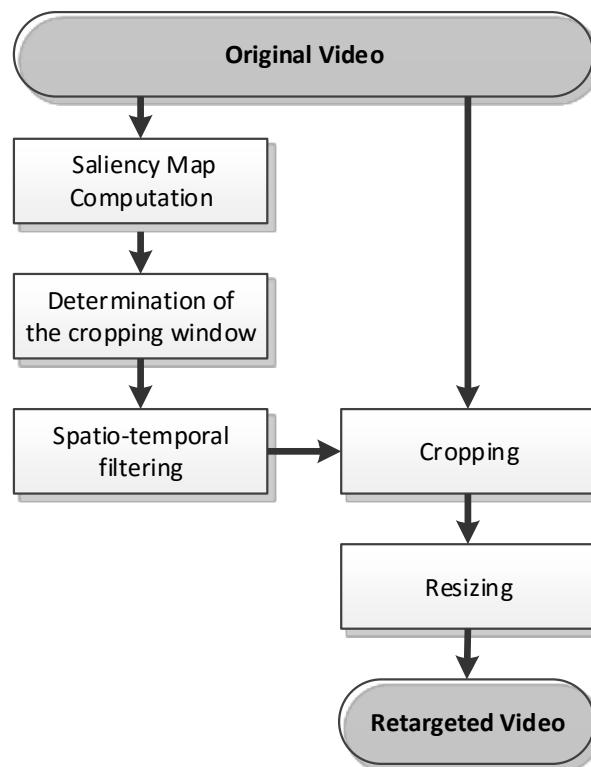


Figure 4.5: Functional diagram of the multi-operator video retargeting method.

### 4.2.1 Determination of the cropping window

After the computation of visual saliency map as described in Section 4.1.1, the identification of the cropping window in the saliency map is performed. The cropping window should guarantee the inclusion the entire important objects i.e., with higher saliency values, inside its perimeter. Towards this end, in the current approach the cropping window size and location are adjusted so that a pre-defined percentage of total energy of the saliency map is included in the crop window. Recalling that the total energy  $E_n$  of the saliency map of frame  $n$  is computed by Equation 4.3,

$$E_n = \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} S_{G_n}(i, j)^2 \quad (4.3)$$

where  $S_{G_n}(i, j)$  is the saliency map value of the frame  $n$  in position  $(i, j)$  and  $W$  and  $H$  are the width and height of the original video sequence, now one wants to determine the values of  $x, y, Cw$  and  $Ch$  such that  $E[c_n(x, y, Cw, Ch)]$  in Equation 4.1 represents a predefined percentage of the total saliency energy  $E_n$ . The method to determine the cropping window is composed of two main steps. The first step consists in the determination of a saliency map  $Current\_S_{G_n}$  which contains only saliency values corresponding a percentage of the total energy of the original  $S_{G_n}$ , for that a threshold-based method was used. In the second step, the cropping window parameters i.e.,  $x, y, Cw$  and  $Ch$ , are determined based on spatial position of the saliency values in the  $Current\_S_{G_n}$ , more details about this method can be found in Algorithm 1.

Figure 4.7 shows a cropping window that includes 70% of the total saliency map energy for a frame of the *Bosphorus* video sequence. In this method, the cropping window size is directly related to the percentage of total energy and so high values of this percentage produce cropping windows with larger size. Conversely, if the percentage is low, then the cropping window size is lower too, and in the limit the cropping window size is equal to the desired retargeted frame size. If the size of the cropping window is larger than the target size, a downsizing step based on a spatial scale change has to be employed to fit the cropped window to the desired value.

**Algorithm 1** Determination of the cropping window

---

```

Set  $Percentage\_E_n$ ;
Compute  $E_n$ ;
 $Perc\_E_n = E_n \times Percentage\_E_n$ ;
 $Saliency\_threshold = 0$ ;
 $Current\_E_n = Energy(S_{G_n})$ ;
 $Current\_S_{G_n}(i, j) = S_{G_n}(i, j)$ ;
 $Previou\_S_{G_n}(i, j) = Current\_S_{G_n}(i, j)$ ;
while  $Perc\_E_n \leq Current\_E_n$  do
     $Saliency\_threshold ++$ ;
    if  $Previou\_S_{G_n}(i, j) \leq Saliency\_threshold$  then
         $Current\_S_{G_n}(i, j) = 0$ 
    end if
     $Previou\_S_{G_n}(i, j) = Current\_S_{G_n}(i, j)$ 
     $Current\_E_n = Energy(Current\_S_{G_n})$ ;
end while
Find coordinates of upper-left corner of the cropping window  $(x, y)$  in  $Current\_S_{G_n}$ ;
Find the size of the cropping window  $(Cw, Ch)$  in  $Current\_S_{G_n}$ ;
Output:  $c_n(x, y, Cw, Ch)$ ;

```

---

### 4.2.2 Spatio-temporal filtering

In order to limit the temporal jitter of the crop window between consecutive frames, a spatio-temporal filtering is used to stabilize both the location and the size of the cropping window. The filter is applied to the crop window parameter set  $C$  which is formed by  $n$  cropping window parameters  $c_n(x, y, Cw, Ch)$ , one for each frame. The filtering procedure followed in the present case is different from the one defined in the previous section, because here both the vectors of position coordinates  $C_x$  and  $C_y$  as well as the crop windows sizes  $C_{Cw}$  and  $C_{Ch}$  are median filtered, as specified in Equation 4.4

$$\begin{aligned}
 \hat{C}_x &= Median((C_x), 15) \\
 \hat{C}_y &= Median((C_y), 15) \\
 \hat{C}_{Cw} &= Median((C_{Cw}), 15) \\
 \hat{C}_{Ch} &= Median((C_{Ch}), 15)
 \end{aligned} \tag{4.4}$$

where  $\hat{C}_x$ ,  $\hat{C}_y$ ,  $\hat{C}_{Cw}$  and  $\hat{C}_{Ch}$  are filtered versions of  $C_x$ ,  $C_y$ ,  $C_{Cw}$  and  $C_{Ch}$  and 15 is the median filter size.



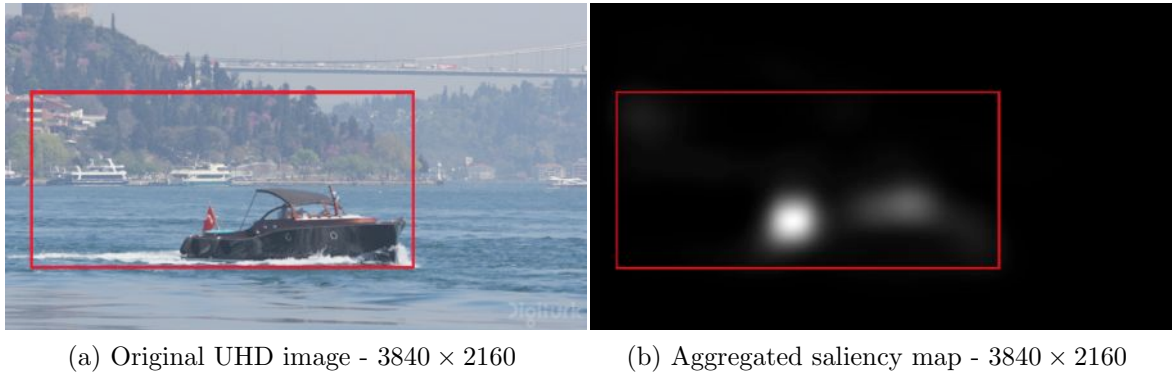


Figure 4.6: Red box is the cropping window with 70% of the total energy - *Bosphorus* video sequence.

### 4.2.3 Cropping

The hybrid retargeting method applies a cropping operation to the original video sequence based on the information of  $\hat{C}$ , obtained in the Section 4.2.2. This operation keeps the visual information within cropping window size ( $\hat{C}w \times \hat{C}h$ ) which represents the most salient information of the original video sequence and discards the information outside the cropping window. The result of this cropping operation for a frame of the *Bosphorus* video sequence is shown in Figure 4.7a.

### 4.2.4 Resizing

Finally, and after the cropping operation, the frame is resizing to the desired resolution ( $W' \times H'$  defined by the user) using as interpolation/downsampling filters those defined and used in the MPEG-4 reference software [186]. Figure 4.7b shows the result of the retargeting process applied to a frame from the *Bosphorus* video sequence, wherein Figures 4.7a and 4.7b show the frame after cropping and downsizing operations, respectively.






Figure 4.7: Result of the application, (a) cropping operation to  $2517 \times 1154$ , (b) downsizing operation to  $1280 \times 720$  - *Bosphorus* sequence.

### 4.3 Results and analysis

The performance of the proposed retargeting methods were evaluated through comparison with the results of three competing methods: direct downsizing method (using the MPEG-4 downsampling filter), centred cropping and seam carving method. A visual comparison was performed as a user-driven approach while the impact of enforcing temporal consistency in the coding efficiency of retargeted video was also evaluated. Rate-Distortion (R-D) efficiency was used for the latter. The *Jockey*, *Bosphorus* and *HoneyBee* UHD test sequences were used in the evaluation. The same HD 720p resolution ( $1280 \times 720$ ) was used for all cases of retargeting from the UHD 4K resolution ( $3840 \times 2160$ ). The original characteristics of each sequence are summarised in Table 4.1.

For the computation of visual saliency maps, the weights  $w_s$ ,  $w_m$  and  $w_f$  of Equation (3.12) of Section 3.1.2 were the same for all frames. Experimentation showed that small variations in these weights do not alter drastically the results and that values  $w_S = w_M = w_f = 1/3$  were a good choice. When face detection is not used the weight  $w_f$  is 0 and the remaining weights are equal to  $w_S = w_M = 1/2$ . The HEVC reference software HM-16.6 was used to encode the retargeted video sequences. The encoder configuration was set to Random Access. The R-D operational points used to obtain the R-D function were obtained from the set of QP =  $\{22, 27, 29, 32\}$  for 100 frames.

Table 4.1: Details of the test sequences used in the experiments.

	Name	#Frames	Resolution
	Jockey	600	3840x2160
	Bosphorus	600	3840x2160
	HoneyBee	600	3840x2160

### 4.3.1 Visual comparison

In order to make a fair visual comparison between the methods in analysis, all test cases used the same ratio for the original and retargeted sequences sizes. Figures 4.8, 4.9 and 4.10 show the visual results of five different retargeting methods: downsizing, centered cropping, seam carving, hybrid and the method presented in Figure 4.1, for the video sequences *Jockey*, *Bosphorus* and *HoneyBee*.

Downsizing preserves the context of the scene but some important objects may not be recognizable due to severe loss of detail. For example in the *Jockey* scene, the horse number and the jockey face are not discernible and in *HoneyBee* sequence the honey bee is almost invisible. In centered cropping it is assumed that the region of interest is always located in the center of the image, but this assumption is not true in many cases, particularly in UHD because the number of regions of interest tend to be spread over larger resolution images. For instance, in Figure 4.8c, one may clearly observe that center cropping is not an acceptable solution. The seam carving method resizes image by preserving the important content and cutting out less important regions while ensuring that the smoothness of the image is retained, i.e., no abrupt cuts are introduced into the image. Nonetheless, when used for retargeting this method has some limitations as shown in Figures 4.8d and 4.9d, where severe geometric distortions are visible in the retargeted frames presented. For example, in Figure 4.9d boat canopy support and flagpole are not straight. Regarding the first method described in this chapter, retargeting with a fixed-size cropping window, the retargeted images contain the essential information of the scene, as shown in Figures 4.8f, 4.9f and 4.10f. Since downsizing is not used, it is possible to keep some high-resolution details, such as the bird on the fence and the face of the rider in the *Jockey* scene or the honey bee in the *HoneyBee* sequence. However, this method has some limitations, since it cuts and ignores important regions and objects especially

when the retargeted frame size is sharply reduced. Thus, to preserve the relevant content of a frame, the hybrid method should be used, with advantages demonstrated in Figure 4.9e that shows a retargeted frame without problematic cuts including the important area of the boat together with some contextual regions which are not preserved when using the first retargeting method proposed in this chapter.

(a) Original -  $3840 \times 2160$ 

(b) Downsize



(c) Centered Cropping



(d) Seam carving [68]



(e) Hybrid proposed



(f) Proposed

Figure 4.8: Visual comparison of retargeted methods of the *Jockey* sequence.



(a) Original -  $3840 \times 2160$ 

(b) Downsize



(c) Centered Cropping



(d) Seam carving [68]



(e) Hybrid proposed



(f) Proposed

Figure 4.9: Visual comparison of retargeted methods of the *Bosphorus* sequence.

(a) Original -  $3840 \times 2160$ 

(b) Downsize



(c) Centered Cropping



(d) Seam carving [68]



(e) Hybrid proposed



(f) Proposed

Figure 4.10: Visual comparison of retargeted methods of the *HoneyBee* sequence.

### 4.3.2 Temporal consistency-visual comparison

In this section, a visual comparison between the results of the first proposed method with and without temporal filtering is presented. In Figures 4.11 and 4.12, the first row shows four consecutive frames of the original sequences and corresponding retargeted frames of the proposed method without and with median filter are shown in second row and third row of the *Jockey* and *Bosphorus* video sequence, respectively.

In the second row of Figure 4.11 is shown an unwanted horizontal displacement between consecutive frames, particularly between the frames of Figure 4.11f and Figure 4.11g. For the case of the *Bosphorus* sequence a vertical displacement is visible between the frames presented in the Figures 4.12e and 4.12f, and also between the frames shown in Figures 4.12f and 4.12g. In third row of Figures 4.11 and 4.12 there are no significant sudden changes in the positions (horizontal and vertical) of the content from frame to frame, i.e., the temporal filtering limits the jitter between adjacent frames. Overall the proposed solution generates temporally smooth retargeted sequences without requiring application of complex motion analysis and compensation methods.

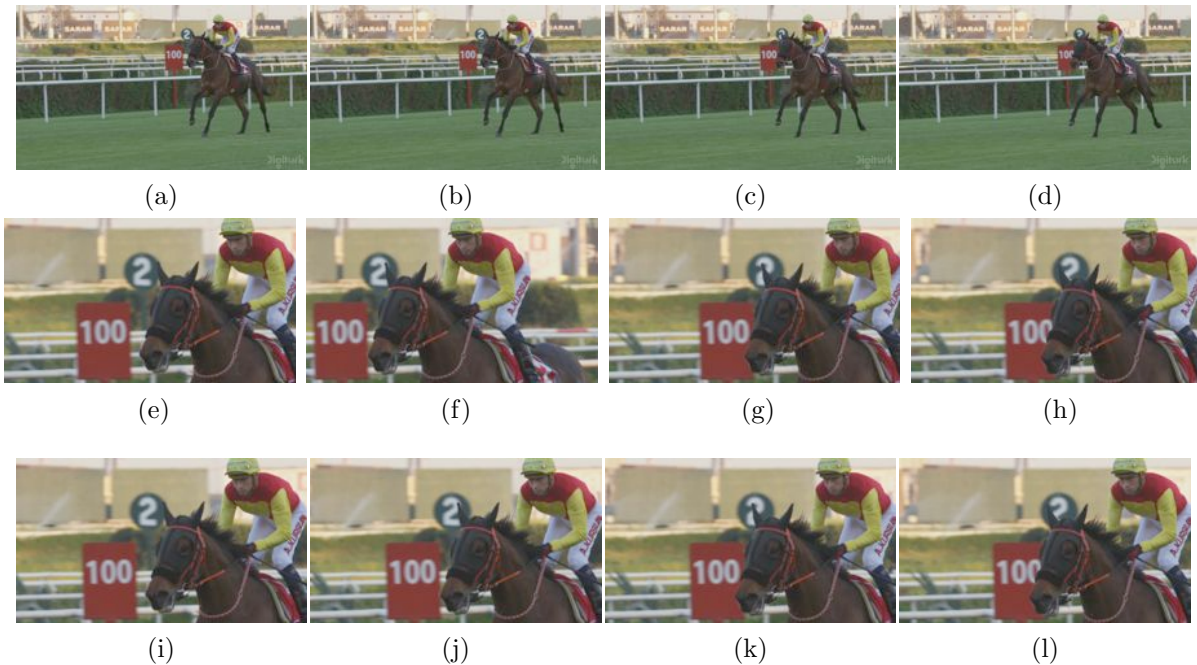


Figure 4.11: (a)-(d) Four consecutive frames of *Jockey* sequence. Retargeted (e)-(h) without median filter and (i)-(l) with median filter.



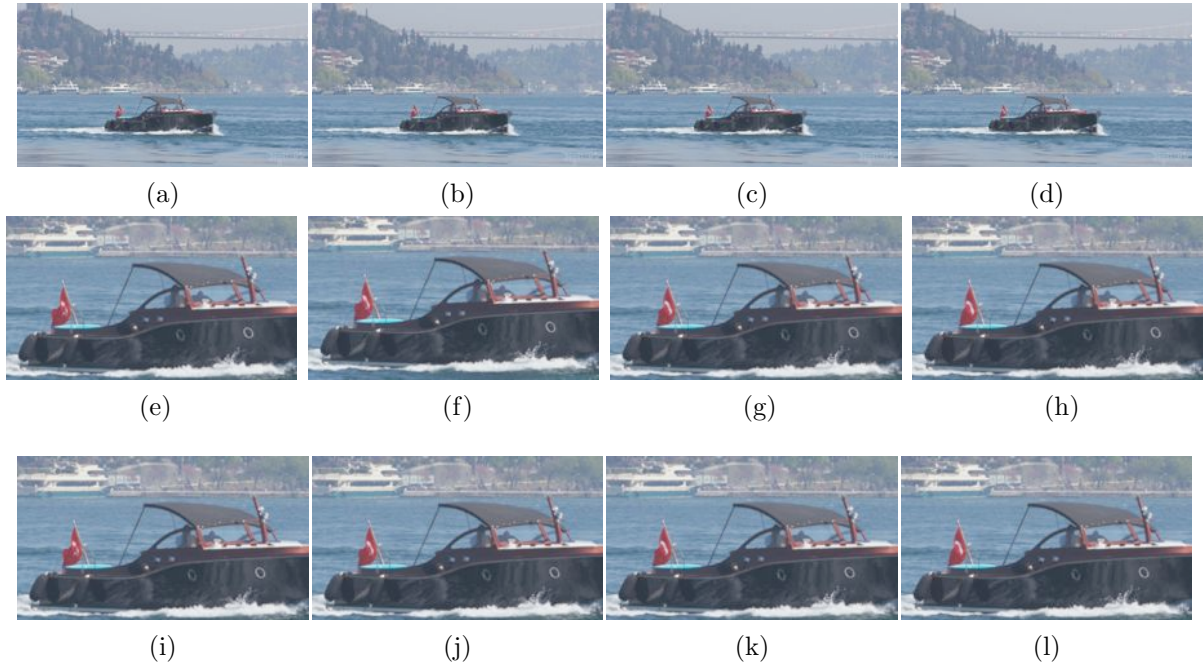


Figure 4.12: (a)-(d) Four consecutive frames of *Bosphorus* sequence. Retargeted (e)-(h) without median filter and (i)-(l) with median filter.

### 4.3.3 The influence of temporal consistency on video encoding efficiency

To evaluate the impact of enforcing temporal consistency in the retargeted video compression efficiency, a set of simulations was carried out specifically for this purpose. In these tests, the R-D efficiency of the retargeted video with and without temporal consistency is compared for three test sequences, *Bosphorus*, *Jockey* and *HoneyBee*. Figures 4.13, 4.14 and 4.15 show that better compression efficiency is always obtained for retargeting with temporal consistency. The difference between the two methods lies in the range of  $0.25 \sim 1.0\text{dB}$  for the three test sequences. This gain results from the larger temporal inter-frame correlation attributable to the temporal consistency improvements. This higher inter-frame correlation leads to a better motion compensated prediction and benefits coding efficiency. When jitter removal is not used and just the maximum-energy cropping window is used to produce the retargeted video, there are sharp discontinuities between the spatial locations of adjacent frames. This leads to non-matching regions in the borders and thus more failures in motion estimation, as well as more complex motion fields which result in lower coding efficiency. Therefore, a smooth trajectory of the

cropping window over the time dimension benefits the R-D efficiency. In the case of the *HoneyBee* sequence (see Figure 4.15), the difference between the two methods is less significant, since the jitter of retargeted video without temporal filter is lower and almost insignificant. From the experiments carried out with various test sequences the maximum observed PSNR gain is approximately of the same magnitude for different types of content, which seems to indicate that more important than the content itself is the jitter removal filter used to enforce the temporal consistency.

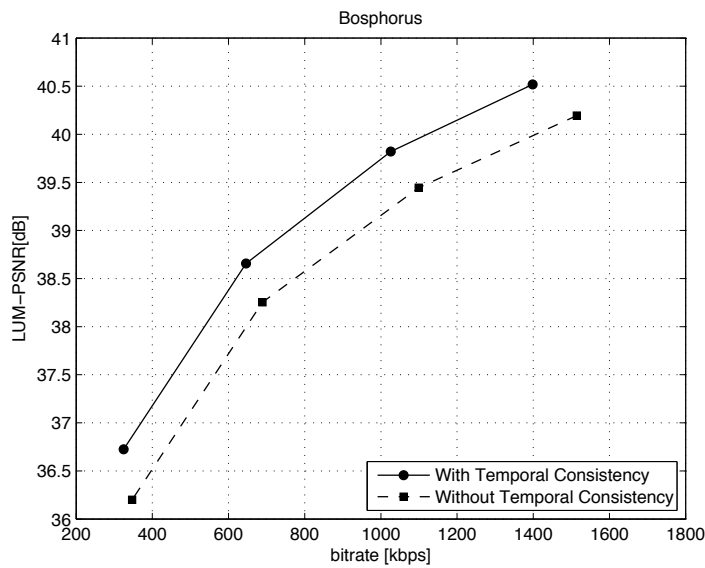
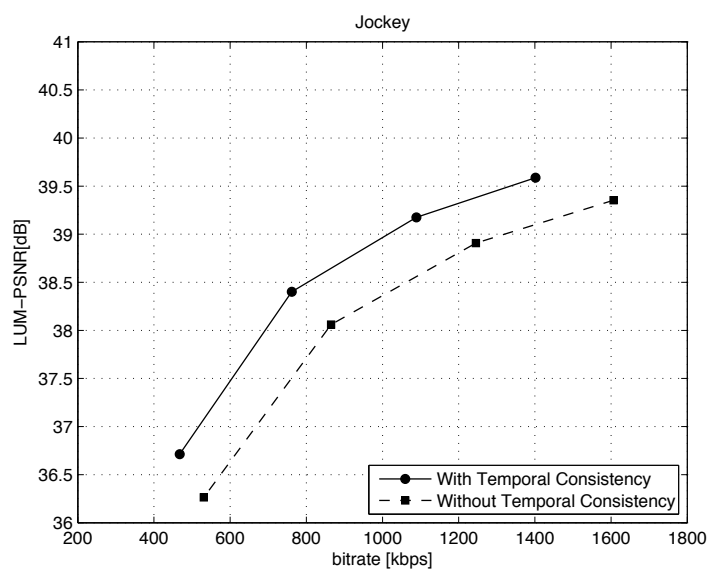
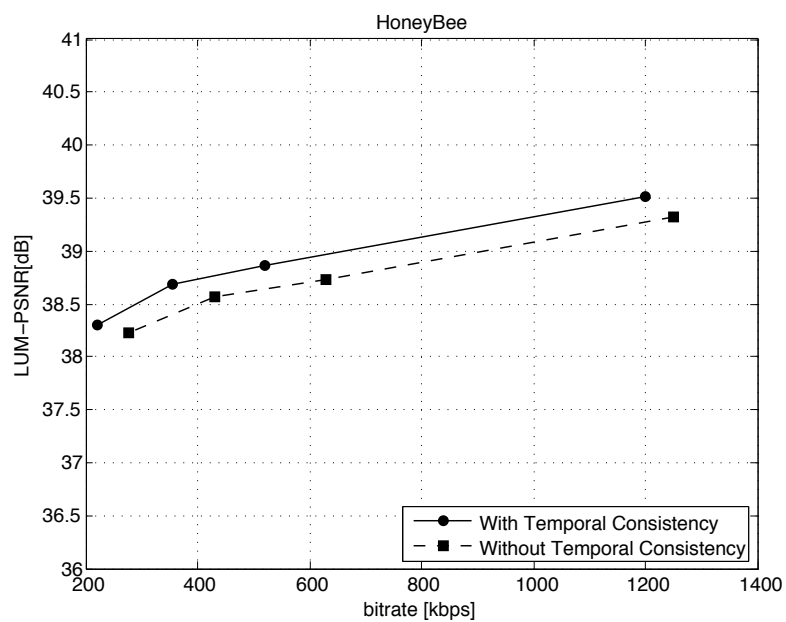


Figure 4.13: R-D of the *Bosphorus* sequence.

Figure 4.14: R-D of the *Jockey* sequence.Figure 4.15: R-D of the *HoneyBee* sequence.

## 4.4 Conclusions

In this chapter video two retargeting methods based on visual saliency maps obtained from visual attention models were presented. A visual comparison of the outputs generated by such method shows their ability to preserve the relevant content of the UHD resolution video. Furthermore, the compression of the retargeted UHD video using HEVC standard is significantly improved when temporal consistency is enforced through filtering for jitter removal, which shows a double benefit, both for the visual quality and for the coding efficiency. Although, the proposed methods are based on visual saliency models it would be important to know how good the methods are in keeping the most interesting regions in the retargeted content. This analysis requires collecting human visual fixation information for different types of content and resolution, a research work which due to its logistic complexity and cost has not been done yet. Another point that deserves further attention, most likely in the near future, is the objective assessment of the proposed retargeting methods in comparison to other similarly aimed competing solutions.

## 3D/2D video summarisation

---

As explained in Chapter 2, most of the video summarisation methods presented in the literature rely on two operations, detection of video shot boundaries and choice of representative frames (key-frames) for each shot. Therefore an entire video sequence is firstly divided into video shots based on scene transitions using one of several SBD methods available in the state-of-the-art and then a key-frame extraction method is applied to each video shot to extract the most representative frames based on specific features of the video. In theory it is possible to dispense with the video shot segmentation step and choose a set of representative frames for the entire video, but in practice this procedure is not practical as it would require storing and analysing an enormous amount of video frames with high memory and computational requirements.

This chapter presents a video shot boundary detection method and four key-frame extraction methods that can be combined to form a framework for the derivation of efficient temporally condensed representations of 2D and 3D video. The framework can be used to create compact versions of the input video sequences, according to meaningful criteria, to ensure that the most relevant visual information of the original sequence is preserved. The shot boundary detection method proposed is based on a clustering technique with only two clusters to ensure computational simplicity and is shown to provide good performance. The first key-frame extraction method described selects a set of frames that is optimal from the point of view of the quality of the video shot reconstructed based on them. This method is not tied to any distortion metric and can be specialized by choosing specific distortion measures. The next method proposed address the complexity issues that ail the previous method by proposing a fast solution. Then the following two methods introduce 3D video specific information and perceptual quality modelling into the summarisation framework by making use of perceptually relevant distortion measures and features derived from the depth of 3D contents in the video summarisation procedures.

The results demonstrate that the proposed methods outperform or achieve similar performance as other summarisation methods. Some of the methods experiments and results presented in this chapter were published in J1, C3, C4, C5, C8 and C9.

## 5.1 Video shot boundary detection

In this section an algorithm for automatic detection of video shots with different perceptual features is presented. The main novel aspects of the proposed algorithm compared to other state-of-the-art algorithms are: (i) the flexibility of the algorithm since that can be applied to different 3D content formats, such as stereoscopic video or video-plus-depth and also to 2D video; (ii) no explicit thresholds are required in the decision processes; (iii) no training is needed.

The proposed shot boundary detection algorithm was initially developed for 3D video, however as a result of its flexible structure which can include different features in the decision process, it can also be applied 2D video shot boundary determination.

In the case of 3D video, the algorithm is able to identify sets of consecutive frames which exhibit depth coherency by using features that capture depth-temporal characteristics. A combination of measures of texture variation along the temporal dimension and depth variance is used in a K-means clustering algorithm to find the 3D video frames which are likely to be true 3DSB. The transitions between 3D video shots can be classified as smooth or sharp according to the speed of change of the visual information over the transitions. They are smooth when gradual transitions occur in both the temporal and depth dimensions, whilst sharp transitions take place when the abrupt temporal transitions and depth discontinuity occur simultaneously. Joint texture variation along the temporal dimension and depth variance (intra-frame) is used by the K-means clustering algorithm to locate 3DSB. The absolute value of frame difference and sum of absolute values of luminance histogram difference are used as dissimilarity metrics in the temporal dimension, while in the depth dimension, the variance of depth of each 3D video frame is used as a measure of depth similarity. The K-means clustering algorithm is used to determine 3DSB frames without resorting to thresholds, nor training sequences to find optimal parameters for decision.

### Proposed 3D video shot boundary detection method

The proposed algorithm relies on depth and temporal information. The temporal information is computed from texture frames, while depth information can be extracted from either depth maps or disparity maps. An adequate combination of these two features in 3D video provides the necessary information for scene shot boundary detection.

The sequence of operations of the proposed algorithm is shown in Figure 5.1. Firstly, the uncompressed 3D video sequence, texture and depth information (disparity or depth maps) are processed to compute the feature vectors with relevant information. In this operation, the temporal variation in image texture and the variance of depth data are computed as the relevant features to be used for 3DSB detection. Since temporal variation is represented by two values (see the algorithm description below), the actual feature vector associated with each 3D frame is composed of three elements, i.e., image difference, histogram difference and depth variance. Secondly, the feature vectors obtained for frame transitions are normalized. Thirdly, a set of 3DSB candidate frames are identified. Next, the K-means clustering algorithm is used to group candidate frames into two clusters, each one containing either 3DSB or non-3DSB frames. The relevant shot boundary transition frames are defined as those located in the resultant 3DSB frames cluster. The clustering process iteratively recomputes the two centroids and re-clusters the candidate frames until convergence is attained and the average Euclidean distance between the feature vectors of candidate frames and the corresponding cluster centroid is minimised. As pointed out previously this algorithm does not incorporate thresholds in the decision process. Furthermore the number of clustering iterations until convergence is small and so its computational complexity is low.

The proposed 3DSB detection algorithm is described in detail as follows:

**Computation of feature vectors** - for each frame  $i$ , the mean of absolute difference  $d_{sad}(i)$  and sum of absolute luminance histogram difference  $d_{hist}(i)$  of adjacent frames are computed according to equations (5.1) and (5.2), respectively. The combination of these metrics provides improved detection accuracy of temporal transitions. In this context,  $d_{sad}(i)$  is a measure of temporal activity, defined as:

$$d_{sad}(i) = \frac{1}{|S|} \sum_{r \in S} |f_{i-1}(r) - f_i(r)| \quad (5.1)$$

where  $f_i(r)$  indicates the pixel value of luminance at spatial position  $r$  in the  $i$ th frame,  $S$  represents the set of pixel positions in the frame, and  $|S|$  is the total number of pixels in a frame.

The sum of absolute difference of the luminance histograms of two consecutive frames provides a low complexity and robust mechanism for measuring temporal activity since this is mostly insensitive to translational, rotational and zooming of camera motions [95].

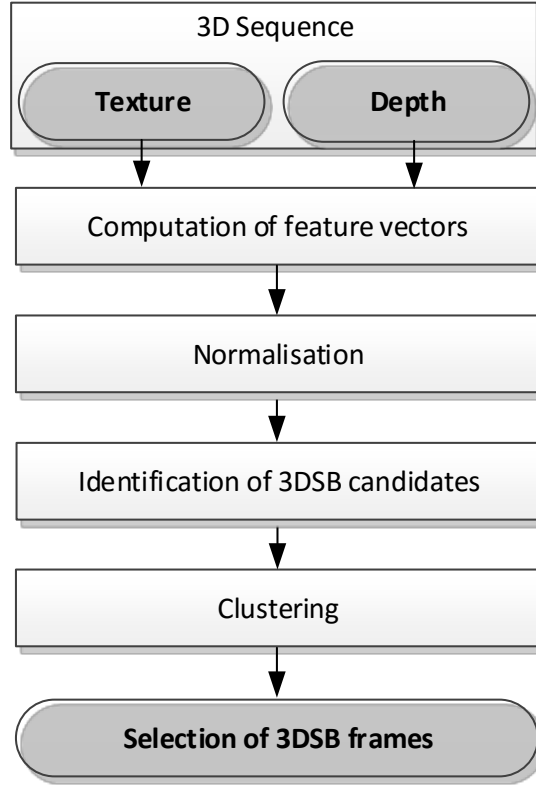


Figure 5.1: 3DSB algorithm architecture.

For each frame  $i$ ,  $d_{hist}(i)$  is calculated by obtaining the histograms of pixel values of the current and previous frames and then computing the sum of the absolute differences of matching bins of the histograms of the two frames. Thus,  $d_{hist}(i)$  is defined as:

$$d_{hist}(i) = \frac{1}{N_b} \sum_{c=1}^{N_b} |H_{i-1}(c) - H_i(c)| \quad (5.2)$$

where  $N_b$  represents the total number of histogram bins, and  $H_i(c)$  represents the number of pixels in frame  $i$  contained in the  $c^{th}$  bin.

For each frame  $i$ , the absolute difference of depth variance  $d_{\sigma_{depth}}(i)$  between frames  $f_{i-1}$  and  $f_i$  is processed according to Equation (5.3).

$$d_{\sigma_{depth}}(i) = |\sigma^2(f_{i-1}) - \sigma^2(f_i)| \quad (5.3)$$

where  $\sigma^2(f_i)$  is the variance of depth in frame  $f_i$ . Using the three scalar features computed as described, a vector of features  $V(i) = [d_{sad}(i), d_{hist}(i), d_{\sigma_{depth}}(i)]$  is defined



for each frame  $i$ .

**Normalisation** - previous to further processing the individual components of the feature vectors of 3DSB candidate frames,  $V^*(i) = [d_{sad}^*(i), d_{hist}^*(i), d_{\sigma_{depth}}^*(i)]$ , are normalised to the range  $[0..1]$ . The feature vector of normalised values for frame  $i$ ,  $V^{\hat{}}(i)$ , is then given by

$$V^{\hat{}}(i) = \left[ \frac{d_{sad}^*(i)}{Md_{sad}}, \frac{d_{hist}^*(i)}{Md_{hist}}, \frac{d_{\sigma_{depth}}^*(i)}{Md_{\sigma_{depth}}} \right] \quad (5.4)$$

where  $Md_{sad}$ ,  $Md_{hist}$  and  $Md_{\sigma_{depth}}$  are the largest possible values of  $d_{sad}^*$ ,  $d_{hist}^*$  and  $d_{\sigma_{depth}}^*$ , respectively. Thus, the range of the feature vectors is a cube with two opposite vertices at  $[0, 0, 0]$  and  $[1, 1, 1]$ . Figure 5.2 shows the feature vectors of all 3DSB candidates frames for the *Knight's Quest 3D* sequence obtained with a temporal window of 5 frames, where x-axis is  $\hat{d}_{sad}$ , y-axis is  $\hat{d}_{hist}$  and z-axis is  $\hat{d}_{\sigma_{depth}}$  of the feature vector normalized  $\hat{V}$ .

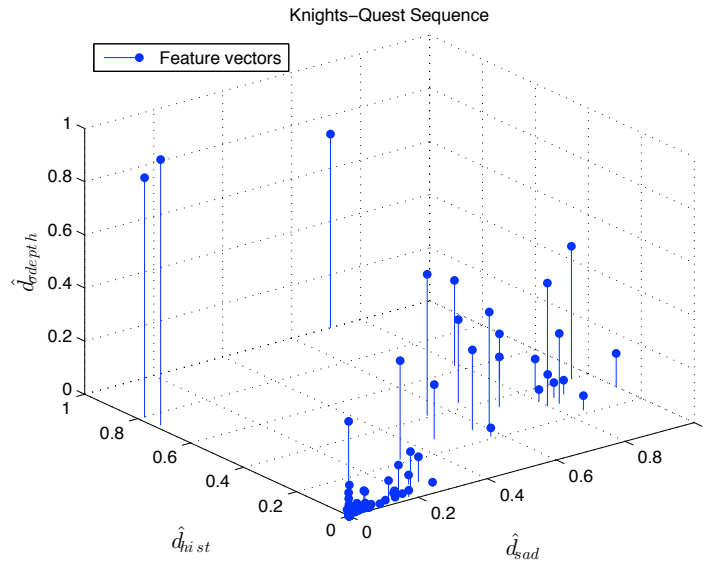


Figure 5.2: Feature vectors of all 3DSB candidates frames.

**Identification of 3DSB candidates** - to reduce the number of initial candidate frames, a pre-selection process is carried out over non-overlapping temporal windows  $W$  with maximum size corresponding to 1 sec. of video, i.e.,  $1 < W_{size} < \#Frames/sec$ . For each window, a maximum of one frame can be selected as 3DSB candidate. Frame  $i$  is

selected as 3DSB candidate, if all three elements of the corresponding feature vector  $V(i)$  are greater than their counterparts within the same window.

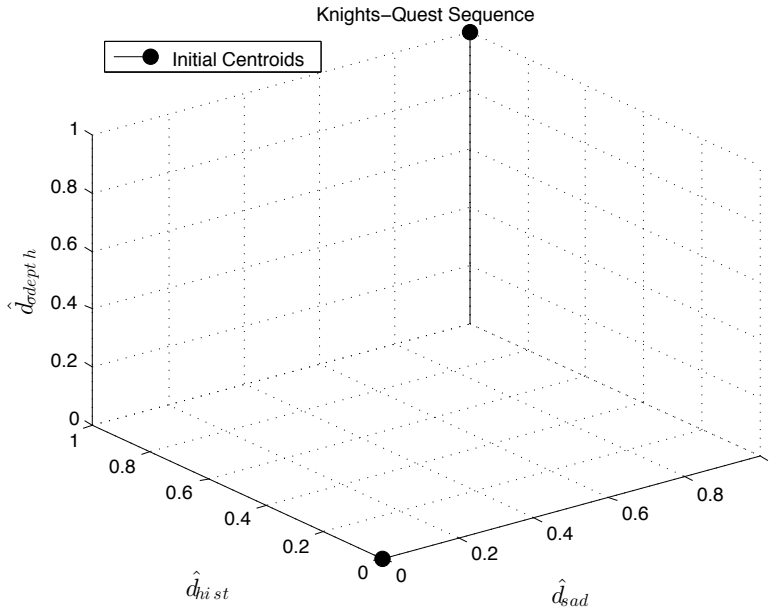


Figure 5.3: Initial centroids of the two clusters.

**Clustering the candidates** - at this stage the K-means algorithm is applied to all 3DSB candidates feature vectors to cluster them into two clusters, i.e.,  $K = 2$ . Figure 5.2 shows the feature vectors of all 3DSB candidates frames of the *Knights Quest 3D* sequence represented in the feature space. The two centroids are initialized to be the largest and smallest possible magnitude vectors  $[1, 1, 1]$  and  $[0, 0, 0]$  as shown in Figure 5.3. The Euclidean distance was chosen as the distance metric to use in the clustering operations. At each iteration, the candidate feature vectors are clustered to the nearest centroid and afterwards the two centroids are recalculated to be the mean vectors of the respective clusters. The process is repeated until the centroids do not suffer any changes. The final clustering distribution puts each feature vector into one of the two clusters and is then used to classify the candidate frames into frames belonging to a shot boundary and other frames.

**Selection of 3DSB frames** - to decide which frames are part of 3DSB it is assumed that the cluster with the largest magnitude centroid is the 3DSB frames cluster. This cluster contains the frames where the relevant 3D transitions occur, according to the three

features represented in the feature vector. Figure 5.4 shows the result of the clustering and selection process of candidates as 3DSB frames of the *Knight's Quest 3D* sequence where the two clusters and respective centroids are clearly identified.

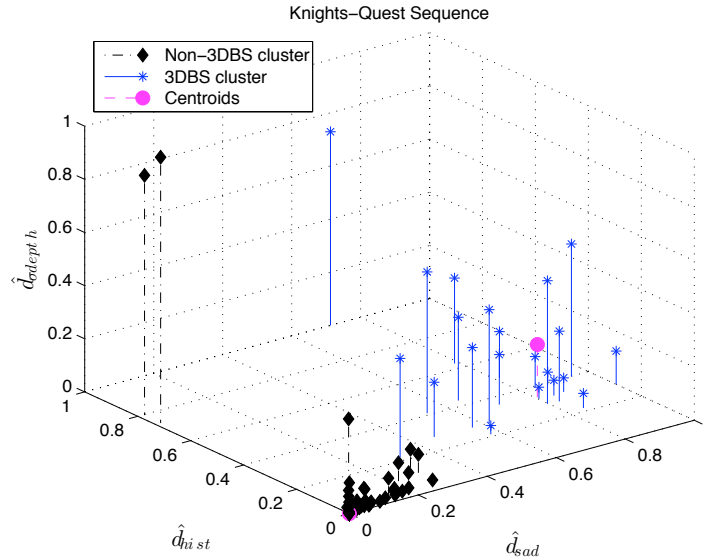


Figure 5.4: Selection of the 3DSB frames.





## Experimental setup and datasets

The performance of the proposed algorithm was evaluated empirically through tests based on four stereoscopic sequences with different duration, resolution, number and type of scene transitions. The characteristics of each sequence are listed in Table 5.1. Sequence *Knight's Quest 3D* is a 3D animation with several sharp and smooth transitions, *Old-timers* shows an outdoor video scene with just a few sharp transitions and sequence *Summer in Heidelberg* shows an elevated view of Heidelberg with several shots connected by smooth transitions. *3DTestSeq* sequence was created by concatenating four sequences: *Ballet*, *BookArrival*, *Kendo* and *Ballons*. This concatenated sequence provides a reference for testing abrupt scene changes. The depth information of the concatenated sequence was obtained by concatenating the original sequences depth maps. All smooth transitions are of type dissolve, fade-in/fade-out, with duration ranging from 20 to 100 frames, approximately.

The disparity maps used in these experiments were computed by the fast bilateral method proposed in [187], with the algorithm parameters set to  $W = 39$ ,  $w = 3$ ,  $\gamma_s = 14$ ,  $\gamma_c = 23$

and TAD-threshold 53. To enable the assessment of the performance of the shot boundary detection algorithm, reference locations for the boundaries of the test 3D videos were marked manually by one observer and confirmed by three other observers. A scene transition was declared to be real when all observers agree on its occurrence. The performance of the proposed 3DSB detection algorithm was measured by the Recall Rate (R), Precision Rate (P) [107] and accuracy measure (F1) [108]. More details of these measures have been reported in Section 2.4.1.

Table 5.1: Details of the test sequences used in the experiments.

	Name	Length	Resolution	#Frames	#Transitions	
					Sharp	Smooth
	Summer in Heidelberg	6:00	1280x720	9000	18	4
	Knight's Quest 3D	1:39	1024x576	2615	21	4
	Oldtimers	0:53	1440x1080	1450	4	0
	3DTestSeq	0:32	1024x768	800	3	0

## Results and analysis

Table 5.2 presents the results obtained by using the proposed algorithm to detect 3DSB in the sequences described above. These results clearly show that high Recall and Precision rates are achieved. It is also clear that sharp transitions are fully detected, as observed for *Oldtimers* and *3DTestSeq* sequences, where F1, Recall and Precision rates reach the maximum possible value. For those sequences with smooth transitions, such as *Summer in Heidelberg* and *Knight's Quest 3D*, the proposed algorithm is not capable of detecting all smooth transitions but the overall detection accuracy is still high. Note that, since smooth transitions are not uniform, i.e., there are different lengths and types, e.g., fades, dissolve, wipe, zoom, rotation, etc, such wide heterogeneity results in increased transition detection difficulty, especially when the elemental types of transitions are combined.

From the experiments, it was found that longer smooth transitions are the most difficult to detect, as expected. It was also found that the accuracy of the algorithm is practically independent of the temporal window size, particularly for small sizes. In the *Knight's Quest 3D* sequence, some dependency was found due to the existence of several sharp

transitions very close to each other. In this case, some transitions are not detected when two or more complete shots are included in one single window. This is an indication that accurate detection of all shot transitions requires small window sizes, e.g., 10 – 15 frames. In regard to depth information, it was found that its quality has significant influence on the algorithm detection accuracy. For example, if a particular frame has an abrupt transition with depth discontinuity, but the algorithm used to compute the disparity map does not provide accurate information for this specific frame, then the 3DSB detection is likely to miss such transition. Figures 5.5 and 5.6 show smooth and sharps transitions of the *Summer in Heidelberg* and *Oldtimers* sequences.

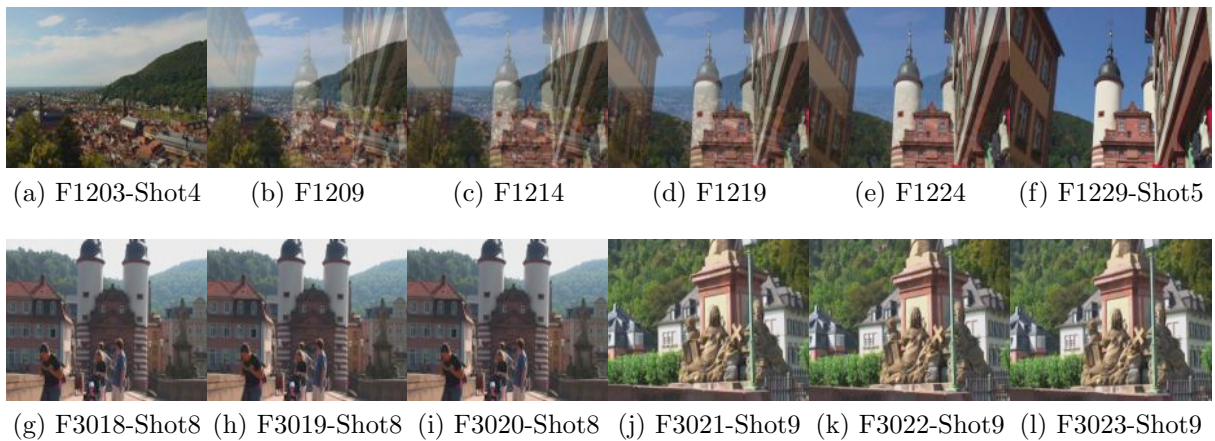


Figure 5.5: *Summer in Heidelberg* sequence: 12 frames corresponding to dissolve smooth transition (a)-(f) and sharp transition (g)-(l).

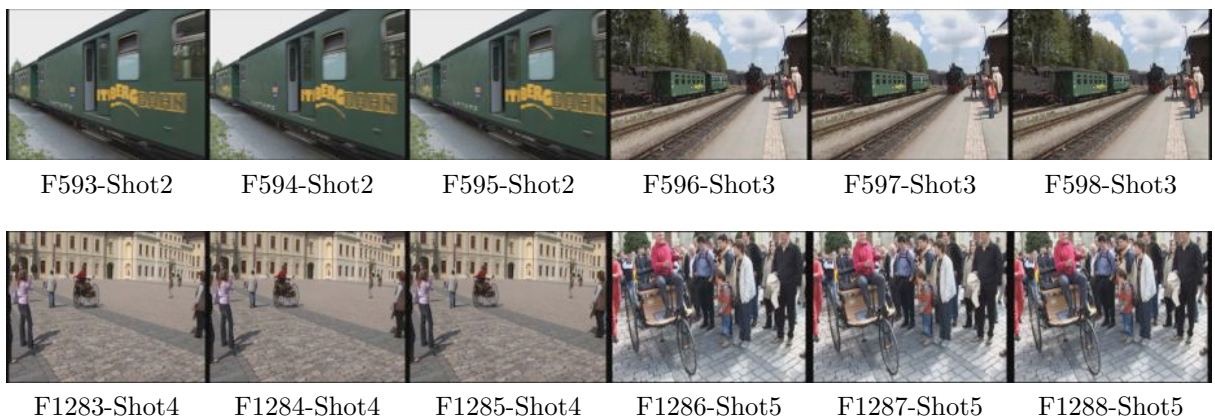


Figure 5.6: *Oldtimers* sequence: 12 frames (3 from each shot) corresponding to 2 sharp transitions.

Table 5.2: Results of the 3DSB detection.

Sequence	Window size	Recall	Precision	F1
Summer in Heidelberg	3	0.95	1	0.97
	5	0.95	1	0.97
	10	0.95	1	0.97
	15	0.95	1	0.97
	20	0.95	1	0.97
Knight's Quest 3D	3	0.95	0.95	0.95
	5	0.95	0.95	0.95
	10	0.95	0.95	0.95
	15	0.90	0.95	0.92
	20	0.90	0.95	0.92
Oldtimers	3	1	1	1
	5	1	1	1
	10	1	1	1
	15	1	1	1
	20	1	1	1
3DTestSeq	3	1	1	1
	5	1	1	1
	10	1	1	1
	15	1	1	1
	20	1	1	1
Average	-	0.97	0.99	0.98

## 5.2 Key-frame extraction methods

In this section, four key-frame extraction methods based on minimum reconstruction error are presented. The first one, is an adaptation of Zhu Li algorithm's [4] and the other three are our proposed ones. A fast solution based on a pre-processing step which reduce computational complexity of the Zhu Li algorithm's and two key-frame extraction methods based on aggregated saliency maps and features derived from the depth for 3D content are proposed and evaluated.

The performance of the proposed methods is evaluated by using different metrics, described in Chapter 2 and comparing them with similar state-of-the-art methods. The results of this evaluation metrics show that proposed methods have a good performance.

### 5.2.1 Optimal key-frame extraction method based on minimum reconstruction error

The key-frame extraction process involves selecting the set of  $m$  key-frames which best represent a temporal shot of  $n$  frames. The number of key-frames  $m$  to be selected can be given as a user-defined parameter or computed according to some predefined criteria. The method used for identification of the representative key-frames is based on the minimisation of the dissimilarity between frames of the original shot and the corresponding ones reconstructed from the set of key-frames. A key-frame set is defined as optimal if a shot reconstructed from these key-frames has the minimum possible distortion (or dissimilarity) when compared to the original shot [4]. In this method, zero-order temporal interpolation is used to reconstruct a shot from a set of key-frames. The key-frames are identified by the temporal indices that indicate their position in the original shot.

#### The problem of optimal key-frame extraction

For convenience, some preliminary concepts and the notation are defined and presented Table 5.3 which are used in the next sections.

Table 5.3: Key-frame summary notation.

Symbol	Description
$F$	Original temporal shot
$n$	Number of frames of the $F$
$K$	Key-frame summary
$m$	Number of frames of the $K$
$F'$	Reconstructed temporal shot

The problem of optimal key-frame extraction can be defined as follows. Let a shot of  $n$  frames be denoted by  $F = \{f_0, f_1, \dots, f_{n-1}\}$ , where the subscripts represent the temporal order of frames. The corresponding set of  $m$  key-frames is denoted by  $K = \{f_{l_0}, f_{l_1}, \dots, f_{l_{m-1}}\}$ , in which  $l_k$  is the frame index (referred to the source set  $F$ ) of the  $k^{th}$  element in  $K$ . Thus,  $K$  is defined by indices  $l_0, l_1, \dots, l_{m-1}$ , such that  $l_0 < l_1 < \dots < l_{m-1}$ . Note that  $l_0, l_1, \dots, l_{m-1}$  correspond to frame indices in shot  $F$  and since in general the key-frames are not equidistant, the  $l_k$  values do not necessarily follow an arithmetic progression.

For example, given a shot  $F = \{f_0, f_1, f_2, f_3, f_4\}$  a possible set of key-frames could be  $K = \{f_0, f_3\}$ , with  $l_0 = 0$  and  $l_1 = 3$ . Then, set  $F' = \{f'_0, f'_1, \dots, f'_4\}$  is reconstructed from the set of key-frames  $K$ , by using zero-order interpolation to fill in missing frames with the most recent one taken from  $K$ , i.e., the same frame is repeated along the time interval where key-frames do not exist, that is,

$$f'_k = f_{i=\max(l)} \quad s.t. \ l \in \{l_0, l_1, \dots, l_{m-1}\}, i \leq k \quad (5.5)$$

Thus, for the previous example the reconstructed set  $F'$  is given by  $F' = \{f_0, f_0, f_0, f_3, f_3\}$ .

The distortion (i.e., dissimilarity)  $D(K)$  associated to the key-frames set  $K$  is computed between the corresponding reconstructed shot  $F'$  and the original one  $F$ , as follows.

$$D(K) = \frac{1}{n} \sum_{k=0}^{n-1} d(f_k, f'_k) \quad (5.6)$$

where  $d(f_k, f'_k)$  is the frame distortion. Note that if  $f_k$  is selected into  $K$ , then  $d(f_k, f'_k) = 0$ . The key-frame ratio  $R(K)$  is defined as the ratio between the number of frames  $m$  in set  $K$  and the total number of frames  $n$  in shot  $F$ , thus,

$$R(K) = \frac{m}{n} \quad (5.7)$$

Using the previous definitions, the key-frames extraction method is formulated as a distortion minimisation problem, where the objective is to find a set of key-frames in each shot that provides the best representation of original temporal shot, under a given maximum key-frame ratio  $R_m$ . Therefore, given the constraint  $R_m$ , the optimum set of key-frames  $K^*$  is the one that minimises the distortion of its corresponding reconstructed shot, i.e.,

$$K^* = \arg \min_K D(K) \quad s.t. \ R(K) \leq R_m \quad (5.8)$$

For example, given a shot  $F$  of  $n = 100$  frames and a key-frame ratio  $R(K) = 0.2$ , the proposed algorithm classifies at most 20 frames as key-frames, i.e.,  $m = 20$ .

### Dynamic programming solution

Assuming that the first frame of any shot is always included in  $K$  there are  $\binom{n-1}{m-1} = \frac{(n-1)!}{(m-1)!(n-m)!}$  different ways to select the key-frames to assemble a rate  $R_m = \frac{m}{n}$  summary.



When  $n$  is large, for typical values of  $m$  smaller than  $n$ , the number of possible solutions is also very large and an exhaustive search for the best solution is not practical. Dynamic programming provides an alternative way to finding a solution to this dissimilarity minimisation problem by breaking it down into simpler subproblems in a recursive manner [188].

A stage  $D_t^k$  is defined as the minimum total distortion incurred by a key-frame set with  $t$  frames, ending at frame  $f_k(l_{t-1} = k)$ . Therefore,

$$D_t^k = \min_{l_1, l_2, \dots, l_{t-2}} \sum_{j=0}^{n-1} d(f_j, f_{i=\max(l)}) \quad s.t. \quad l \in \{l_1, l_2, \dots, l_{t-2}\}, i \leq j \quad (5.9)$$

Note, that  $l_0 = 0$  and  $l_{t-1} = k$  are removed from the optimisation process and  $0 < l_1 < l_2 < \dots < l_{t-2} < k$ , and  $i \leq j$ . After some manipulation, the above formulation for stage  $D_t^k$  can be broken into two parts (see Equation (5.10)), where the first part is the previous distortion stage  $D_{t-1}^{l_{t-2}}$  already computed, (i.e., it represents the minimum total dissimilarity produced by the set with  $t - 1$  key-frames, ending at frame index  $l_{t-2}$ ) and the second part,  $e^{l_{t-2}, k}$  represents the distortion reduction, when frame  $k$  is selected into the set of  $t - 1$  key-frames ending at frame  $l_{t-2}$ . This leads to the following Equation (5.10),

$$D_t^k = \min_{l_{t-2}} \{D_{t-1}^{l_{t-2}} - e^{l_{t-2}, k}\} \quad (5.10)$$

where the distortion reduction is defined as,

$$e^{l_{t-2}, k} = \sum_{j=k}^{n-1} [d(f_j, f_{l_{t-2}}) - d(f_j, f_k)] \quad (5.11)$$

Since the first frame of shot  $F$  is always selected into set  $K$ , the initial distortion stage  $D_1^0$  is given as

$$D_1^0 = \frac{1}{n} \sum_{j=1}^{n-1} d(f_0, f_j) \quad (5.12)$$

Given the above equations, it is possible to compute the distortion stage  $D_t^k$  for any set of  $t$  key-frames ending at frame  $k$  by the recursion defined in Equation (5.10) using the initial distortion stage Equation (5.12).

The optimal set of  $K^*$  key-frames is given by computing the frame indices  $l_0, l_1, \dots, l_{m-1}$

found according to:

$$\begin{aligned}
 l_{m-1} &= \arg \min_c \{D_m^c\} \quad c \in \{m-1, m, \dots, n-1\} \\
 l_t &= \arg \min_{l_t} \{D_{t+1}^{l_t} - e^{l_t, l_{t+1}}\} \quad t \in \{1, 2, \dots, m-2\} \\
 l_0 &= 0
 \end{aligned} \tag{5.13}$$

where  $l_0$  and  $l_{m-1}$  represent the first and last frame index selected into set  $K^* = \{f_{l_0}, f_{l_1}, \dots, f_{l_{m-1}}\}$ , respectively.

The evaluation of the key-frame extraction method based on minimum reconstruction error is performed in the next sections with specific dissimilarity (distortion) metric  $d(f_k, f'_k)$ .

## 5.2.2 Fast key-frame extraction method based on MSE and PCA

This section describes a fast key-frame extraction method based on a pre-processing step that reduces the computational complexity of the solution presented in Section 5.2.1. Both methods use dynamic programming to find the key-frame sets that are optimal in a temporal rate-distortion sense. The methods described use two distortion metrics, one based on a simple MSE measure and the other making use of Principal Components Analysis (PCA). The performance of the key-frame extraction methods are compared in terms of the computational complexity and selected key-frame quality as measured by the reconstructed video distortion. The key-frame extraction methods are then compared from the perspective of trade-off between the distortion and the computation complexity. To ease the understanding of the methods, the definitions and formulations used in their description are presented next.

### Definitions and formulations

**Frame distortion** - the distortion is measured by the distance between two frames  $f_j$  and  $f_k$  and is denoted by  $d(f_j, f_k)$ . Different metrics can be used to calculate this frame distortion  $d(\cdot)$ . In this study MSE and a metric based on a PCA decomposition are used.

The MSE metric is given by:

$$d(f_j, f_k)_{MSE} = \frac{1}{h \times w} \sum_{y=0}^{h-1} \sum_{x=0}^{w-1} (f_j(x, y) - f_k(x, y))^2 \quad (5.14)$$

where  $h \times w$  is the frame size. The PCA metric is the Euclidean distance between two frames in PCA space. The PCA metric is defined as:

$$d(f_j, f_k)_{PCA} = \sqrt{\|T(S(f_j)) - T(S(f_k))\|^2} \quad (5.15)$$

where  $S(\cdot)$  denotes a spatial downsampling process applied to the original frames to reduce their resolution (using the MPEG-4 downsampling filter [186]) and  $T$  is the PCA transform [4].

**Frame-by-frame distortion** - this distortion is denoted by  $d(f_k, f_{k-1})$  and it is a metric that reflects the temporal activity of the video sequence. It is a specific case of the frame distortion defined before, computed between the current frame  $f_k$  and the immediately preceding frame  $f_{k-1}$ . The basis distance function  $d(\cdot)$  can also be MSE or PCA as defined above.

### Proposed method

The fast key-frame extraction method proposed here is based on a simple principle which is confirmed by observation of optimal summaries, the temporal density of key-frames tends to be higher in segments (temporal shots) with higher temporal activity. Therefore an estimate of the temporal activity obtained using the frame-by-frame distortion introduced in the previous paragraphs, can be used to decide how many key-frames should be used to represent a video sub-segment or conversely how long should a sub-segment be to be properly represented by a given number of key-frames. Figure 5.7 shows a plot of the frame-by-frame distortion metric computed on the *Foreman* sequence where are clearly visible high activity segments like that from frame 270 to frame 330 and low activity segments like that from frame 350 to 400. A possible way to use this activity information to accelerate the summarisation is to first split the original video/temporal shot into sub-segments with same temporal activity and then independently summarise each of these sub-segments. Since the dynamic programming algorithm used to extract the key-frames has better performance, in terms of processing time, for small search windows,

this split-and-summarise two step procedure reduces the computational complexity when compared to a full segment summarisation. The algorithm of the method proposed works

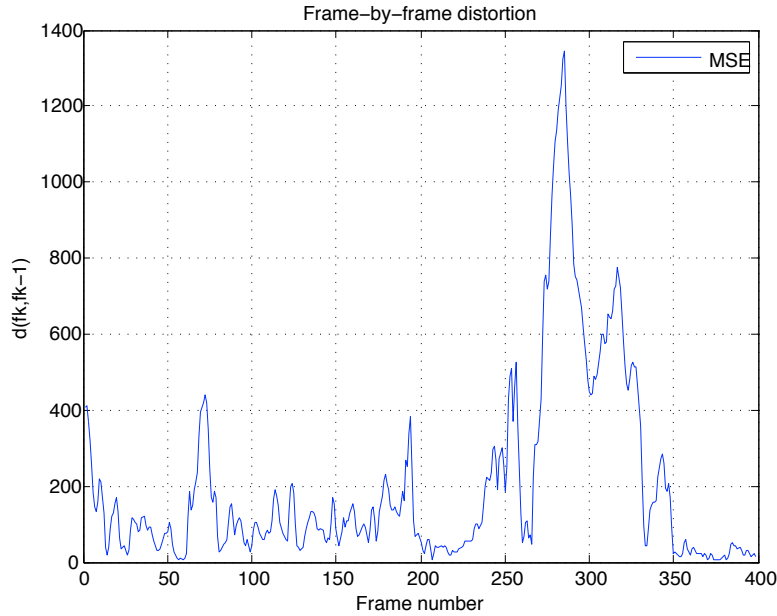


Figure 5.7: Frame-by-frame distortion of the *Foreman* sequence.

as follows; first an average sub-segment activity/distortion  $AD(S)$  is computed, according to Equation (5.16), which requires that the number of desired sub-segments  $n_{seg}$  has been defined beforehand, i.e., the number of sub-segments is a user-defined parameter.

$$AD(S) = \frac{1}{n_{seg}} \sum_{k=1}^{n-1} d(f_k, f_{k-1}) \quad (5.16)$$

Then a scanning of the original video/temporal shot to be partitioned is started, where frames are sequentially assigned to each sub-segment starting with the first, until each sub-segment has an accumulated distortion equal to  $AD(S)$  and all frames have been processed, as detailed in Algorithm 2.

Once the number of frames in each sub-segment is known and since we want that all sub-segments are summarised with the same temporal rate  $R(K)$ , the number of key-frames in a given sub-segment,  $m$ , is computed multiplying the total number of frames in that sub-segment  $n$  by  $R(K)$ , i.e.,  $m = R(K) \times n$ . For instance if we desire a constant  $R(K) = 0.40$  for all sub-segments and one sub-segment has  $n = 27$  frames, it will be represented using  $m = 11$  key-frames.

---

**Algorithm 2** Video segment splitting algorithm

---

```

Input Definition: number_of_sub_segments
Compute:  $AD(S)$ 
 $FrameIdx = FirstFrameIdx$ 
for  $SubSegmentIdx = 1$  till  $number\_of\_sub\_segments$  do
  Set to Zero:  $current\_sub\_segment\_distortion$ 
  Set to Zero:  $CountTotalFramesInSubSegment\_i$ 
  while  $current\_sub\_segment\_distortion \leq AD(S)$  do
    Add  $FrameIdx$  to  $CurrentSubSegmentList$ 
    Increment:  $FrameIdx$ 
    Increment:  $FramesInSubSegment\_i$ 
    Update:  $current\_sub\_segment\_distortion$ 
  end while
  Increment:  $SubSegmentIdx$ 
end for

```



---

The next step uses the key-frame extraction method presented in the Section 5.2.1 to summarise each sub-segment extracting the desired number of key-frames as computed according to  $R(K)$  and the length of the sub-segment. In the experiments performed, both the MSE and PCA defined in Equations (5.14) and (5.15) were used as distortion measures  $d(f_k, f'_k)$  during the summarisation.

**Results and analysis**

A set of experiments was conducted to evaluate the performance of the method proposed in this section. The performance was measured in terms of computational complexity and distortion of the reconstructed video distortion. Those two sets of values (complexity and distortion) were compared to the corresponding data obtained when the algorithm of 5.2.1 was used to summarise the entire video shot without splitting into sub-segments. Additionally, a computational complexity comparison between PCA and MSE distortion metrics, used in key-frame extraction method, was done and the results presented. In the context of this Thesis, the computational complexity is equated with the time spent to construct a key-frame summary and is measured in seconds [s]. The experiments were performed on a desktop computer with a 2.4GHz processor and 1.0GB of RAM memory. In all simulations the temporal rate used was  $R(K) = 0.4$  (good trade-off between distortion and conciseness of the key-frame summary). The video sequences *Foreman* and *Mother&Daughter* were used, both with QCIF resolution at 30 frames per second. The characteristics of each sequence are listed in Table 5.4.

Table 5.4: Details of the test sequences.

	Name	#Frames	Resolution
	Foreman	100	176x144
	Mother&Daughter	100	176x144

**Performance of the fast key-frame extraction method** - the results of the fast key-frame extraction method and Zhu Li's algorithm using the MSE and PCA metrics are shown in Tables 5.5 and 5.6 for *Foreman* and *Mother&Daughter* sequences, respectively.

Each table shows values for the computational complexity and distortion (which is calculated with the Equation (5.6)), for different  $n_{seg}$  (1- Zhu Li's method, 3, 4 and 5). As can be observed from the data in the Tables 5.5 and 5.6 increasing the number of sub-segments results in a decrease of computation complexity with a slight increase in distortion. The reduction of computational complexity is independent of the video sequence and distortion metric chosen. These results were expected as the division of the video sequence/temporal shot into sub-segments decreases the size of the search windows used by the method and the key-frame summary is easier to build. The distortion values (MSE and PCA) vary over different range of values for the two test sequences. The reason of this fact is explained by type and magnitude of the motion present in the test sequences. While the *Mother&Daughter* sequence is characterized by slow motion, the *Foreman* is a medium-speed motion sequence. The best result for *Foreman* (Table 5.5) is obtained when the original sequence is divided in three segments ( $n_{seg} = 3$ ) here, the computational complexity is 13% and 15% of the complexity of summarising the entire sequence (Zhu Li's algorithm) for the two metrics (MSE and PCA) and the distortion is approximately the same for both metrics. In the case of the *Mother&Daughter* sequence (Table 5.6), the results are quite similar to those of the *Foreman* sequence.

In conclusion, the proposed method which pre-partitions the temporal shot to be summarised into sub-segments reflecting the temporal activity distribution does indeed reduce the computational complexity of the key-frame extraction method presented in Section 5.2.1, with gains of nearly 85% at about the same distortion, as demonstrated using two test sequences.

Table 5.5: Performance of the proposed method *vs* Zhu Li’s [4] method for *Foreman* sequence.

Methods	Metric	MSE		PCA	
	$n_{seg}$	Computational complexity[s]	Distortion MSE	Computational complexity[s]	Distortion PCA
Zhu Li [4]	1	478.00	81.72	273.60	1.43
	3	62.12	82.68	41.70	1.44
Proposed	4	18.06	86.35	20.30	1.46
	5	11.23	86.97	15.20	1.49

Table 5.6: Performance of the proposed method *vs* Zhu Li’s [4] method for *Mother&Daughter* sequence.

Methods	Metric	MSE		PCA	
	$n_{seg}$	Computational complexity[s]	Distortion MSE	Computational complexity[s]	Distortion PCA
Zhu Li [4]	1	496.32	6.18	276.77	0.20
	3	62.73	7.14	42.20	0.21
Proposed	4	18.17	10.81	20.50	0.24
	5	11.30	11.43	15.62	0.27

**Distortion metrics comparison** - the computational complexities incurred summarising video sequences *Foreman* and *Mother&Daughter* are listed in Tables 5.7 and 5.8. It is noticeable that computation complexity increases when the relation  $n - m$  increases for both the MSE and PCA distortion metrics, for both test sequences. Overall, the PCA metric results in lower-complexity than MSE, but for small values of  $n$  (e.g.,  $n = 20$ ) the MSE metric is faster. The PCA metric is faster than MSE, because the resolution of the sequence on which the PCA measure is computed is first reduced through downsampling, after which the PCA transform is applied. For a sequence with QCIF resolution the PCA transform is applied to a downsampled video with resolution 8x6 and while the MSE is computed on the 176x144 resolution original video. When these metrics are computed with images of the same resolution, the processing time of the PCA is higher than the MSE metric.

To verify if the use of one or the other metric influenced the identity of the frames selected for the summary, the index of the key-frames obtained using the two distortion metrics (MSE and PCA) as well as and the number of key-frames common to the two summaries are listed in Tables 5.9 and 5.10 for sequences *Foreman* and *Mother&Daughter*, respectively. On average the two summaries have about 49% key-frames in common for the *Foreman* sequence and 51% for the *Mother&Daughter* sequence.

In the next section, a key-frame extraction method for 3D video is proposed.

Table 5.7: Computational complexity of the *Foreman* sequence.

$n$	$m$	$R(K)$	MSE[s]	PCA[s]
20	8	0.4	0.75	3.78
40	16	0.4	12.16	6.48
60	24	0.4	61.73	42.01
80	32	0.4	194.61	117.36
100	40	0.4	478.00	273.60

Table 5.8: Computational complexity of the *Mother&Daughter* sequence.

$n$	$m$	$R(K)$	MSE[s]	PCA[s]
20	8	0.4	0.83	3.76
40	16	0.4	12.56	6.53
60	24	0.4	63.66	42.30
80	32	0.4	212.50	118.41
100	40	0.4	496.32	276.77

Table 5.9: Key-frames of the *Foreman* sequence.

$n$	$m$	$R(K)$	Key-frames index MSE distortion metric	Key-frames index PCA distortion metric	# common Key-frames index
20	8	0.4	0,2,4,6,8,10,12,17	0,3,5,6,10,12,14,18	3
40	16	0.4	0,2,4,6,8,10,12,16, 18,20,22,25,29,32,35,37	0,3,5,6,10,12,14,18, 20,21,24,25,29,30,32,34	9
60	24	0.4	0,1,2,4,6,8,10,12,15, 17,19,21,24,28,30,32,35,37, 40,43,47,50,52,55	0,3,6,10,12,14,18,20,24, 25,29,30,32,34,35,38,42,44, 46,47,49,51,54,57	11
80	32	0.4	0,2,4,6,8,10,12,15,17, 19,21,24,28,30,32,35,37,40, 43,47,50,52,55,63,65,67,69, 71,72,74,76,78	0,3,6,10,12,18,20,24,25, 29,30,32,34,35,38,42,44,46, 47,49,51,54,57,60,62,63,65, 68,73,74,76,79	16
100	40	0.4	0,2,4,6,8,10,12,15,17, 19,21,24,28,30,32,35,37,40, 43,47,50,52,55,63,65,67,69, 71,72 73,74,76,78,80,84,87, 89,92,94,97	0,3,6,10,12,14,18,20,24, 25,29,30,32,34,35,38,42,44, 46,47,49,51,54,57,60,62,63, 65,68,73,74,76,77,78,80,86, 88,93,97,98	21



Table 5.10: Key-frames of the *Mother&Daughter* sequence.

$n$	$m$	$R(K)$	Key-frames index MSE distortion metric	Key-frames index PCA distortion metric	# common Key-frames index
20	8	0.4	0,1,3,5,7,9,14,17	0,1,3,4,7,8,9,12	5
40	16	0.4	0,1,6,9,12,20,24,26, 27,28,29,30,31,33,35,37	0,3,4,6,21,22,23,24, 26,28,29,30,32,34,36,38	7
60	24	0.4	0,12,20,24,26,27,28,29,30, 32,34,37,40,43,45,47,48,49, 51,53,55,57,58,59	0,3,4,6,22,23,24,26,28, 30,32,34,39,41,43,45,46,47, 48,50,52,57,58,59	14
80	32	0.4	0,12,20,25,27,29,31,33,36, 39,43,45,47,49,51,53,55,57, 58,59,60,61,62,63,64,65,67, 69,71,74,76,78	0,3,4,6,22,23,26,28,30, 32,34,39,41,43,45,46,47,48, 50,52,57,58,59,60,61,62,63, 65,68,74,77,79	14
100	40	0.4	0,12,20,25,27,29,31,33,37, 41,44,46,48,50,52,54,57,58, 59,60,61,62,63,64,65,67,69, 72,75,77,79,81,84,86,88,90, 92,94,96,98	0,3,4,6,22,23,26,28,30, 34,41,43,45,46,47,48,50,52, 57,58,59,60,61,62,63,65,68, 74,80,82,83,84,85,87,89,90, 93,96,97,98	18

### 5.2.3 3D key-frame extraction based on perceptually relevant depth regions

To the best of the authors' knowledge, there is no generic solution for the problem of 3D video summarisation, be it stereoscopic or video-plus-depth 3D video. Most of the existing summarisation methods for 3D video were developed for use with 3D mesh models and so are unsuitable for others 3D formats. A further problem is the lack of common performance evaluation frameworks which makes comparing the few existing 3D video summarisation methods a difficult or even impossible task.

This section describes a method to select the most representative 3D key-frames from 3D video sequences to build a 3D video summary. The method is based on the algorithm of Section 5.2.1 combined with a 3D specific dissimilarity measure that considers the relevancy of the texture and depth information of the frames under consideration for inclusion in the summary.

The method proposed first divides the original 3D video sequence into temporal segments, i.e., 3D video shots, using an algorithm based on clustering of depth-temporal features and derived from the one described in Section 5.1. Next, for each video shot a set frames is automatic selected, where the number of frames chosen per shot is based on some predefined criteria, such as the scene motion or is given as a user-defined parameter.

Overall, the main novel aspects of the proposed method are:(i) the flexibility of the method

since it can be applied to different 3D video formats, such as stereoscopic video and video-plus-depth; (ii) the distortion measure used in the key-frame selection takes into account the perceptual relevancy of the depth information; (iii) better performance than UnS and 2D summarisation method based on clustering (Clu).

### Proposed method

The functional diagram of the key-frame extraction method for 3D video (stereo or video-plus-depth) is shown in Figure 5.8. First, the 3D video is fragmented into video shots based on depth-temporal feature with a specific SBD algorithm. This algorithm combines three visual features (color histogram, pixels difference and depth relevance regions area) and uses K-means clustering to detect video transitions. After that, an key-frame extraction method is used to select the most the 3D frames in each video shot to build the desired 3D key-frame summary. In the Figure 5.8,  $m$  is the number of key-frames and it also defines key-frame summary size.

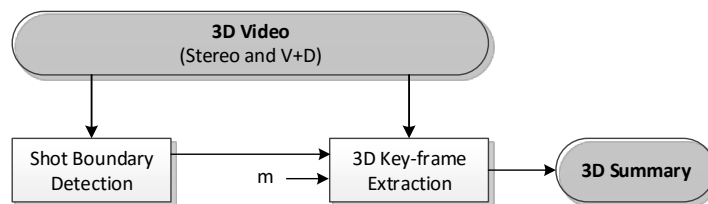


Figure 5.8: Functional diagram of the 3D key-frame extraction method based on perceptually relevant depth regions.

Now, it is explained how the perceptual relevancy of the depth is quantified, for use in the key-frame extraction procedure.

**Depth relevance regions identification method** - Depth Sense Metric (DSM) is a measure of the depth similarity of two 3D frames  $f_k$  and  $f_n$ . DSM is based on the principle that depth information of image regions around scene objects is perceptually more important than that of other regions. Therefore the depth information of these regions has to have a higher weight when measuring the quality or relevance of 3D video frames, as shown in [189], and [190, 191].

Since depth information can be present in explicit or implicit form in 3D video, as is the case in video-plus-depth and stereo video, respectively, the computation of the depth relevancy depends on the 3D video format chosen.

In the next subsections two methods for computing this depth relevancy measure, one for each 3D video format, will be explained in detail.

**Stereo Video Relevancy Region Computation** - for this 3D video format, the method to identify the depth relevance regions based on [189] proceeds as follows:

1. Compute the absolute difference  $AD_{f_k}(x, y)$  between the two views of the stereo frame  $f_k$  under consideration, according to Equation (5.17).

$$AD_{f_k}(x, y) = |fl_k(x, y) - fr_k(x, y)| \quad (5.17)$$

$$0 \leq x \leq M-1; 0 \leq y \leq N-1$$

where  $fl_k(x, y)$ ,  $fr_k(x, y)$  are the left and right views of stereo frame  $f_k$ , respectively and  $M \times N$  is frame size.

2. Remove noise from  $AD_{f_k}(x, y)$  using a median filter ( $3 \times 3$ ). The low magnitude elements are suppressed based on the principle that their relevancy for depth perception is also low.
3. Repeat previous two steps for stereo frame  $f_n$ .
4. Based on the results from the previous steps, determine binary masks  $M(f_k)$  and  $M(f_n)$  for the two stereo frames under comparison  $n$  and  $k$ , by setting to 255 all points of the filtered  $AD(\cdot)$  located in pixel positions where the difference is non-zero and setting to zero otherwise.
5. Compute the intersection mask  $IM(f_k, f_n)$  as the intersection of the nonzero points in masks  $M(f_k)$  and  $M(f_n)$  as given below,

$$IM(f_k, f_n) = M(f_k) \cap M(f_n) \quad f_k \neq f_n \quad (5.18)$$

This step is common to the video-plus-depth format. The intersection mask  $IM(f_k, f_n)$  defines the points where the DSM is computed.

The results of the above method are shown in Figure 5.9 for the *Pantonime* sequence. Figure 5.9c shows  $AD_{f_k}(x, y)$  of the left (Figure 5.9a) and right (Figure 5.9b) views of the first frame  $f_0$ . Figures 5.9d and 5.9e shows the depth relevance regions (i.e., binary masks) of frames  $f_0$  and  $f_1$ , respectively. The intersection mask  $IM(f_0, f_1)$  is presented in Figure 5.9f.

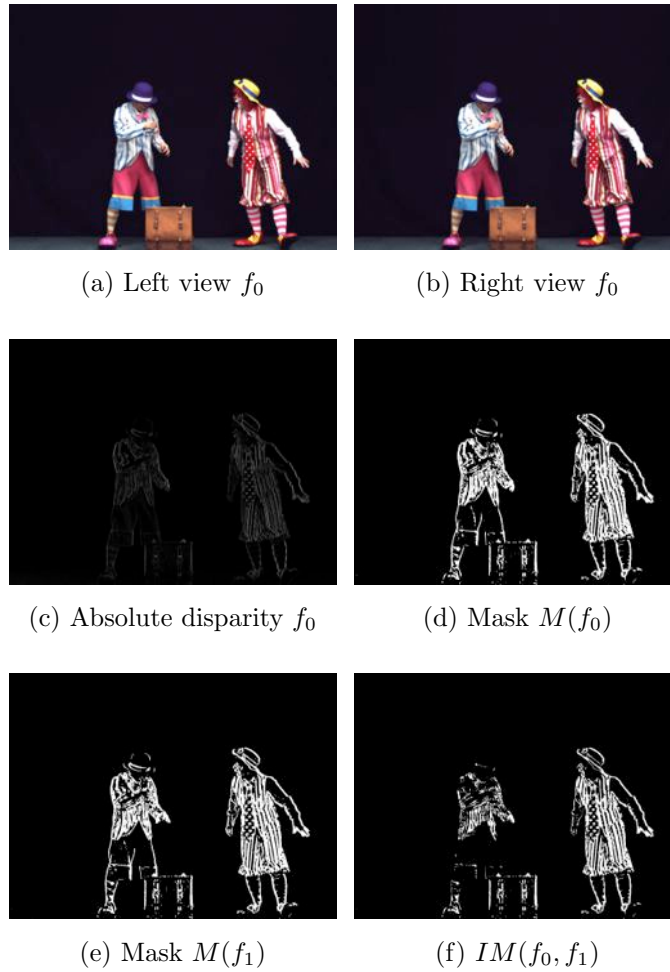


Figure 5.9: Steps for the calculation depth relevance region of the *Pantomime* sequence.

**Video-plus-depth** - for this 3D video format, the depth relevance regions are found through the following 4-step procedure:

1. Extract the edge of the depth map, by the application of the Canny edge algorithm [192].
2. Each edge pixel is then “dilated”, using as structuring element a disk shape with radius equal to 10 pixels.
3. Based on the two previous steps, determine binary masks  $M(f_k)$  and  $M(f_n)$  for the two frames under comparison  $f_n$  and  $f_k$ , by setting to 255 all pixel positions where the filtered edge map is non-zero and setting to zero otherwise.
4. Compute the intersection mask  $IM(f_k, f_n)$ , according to Equation (5.18).

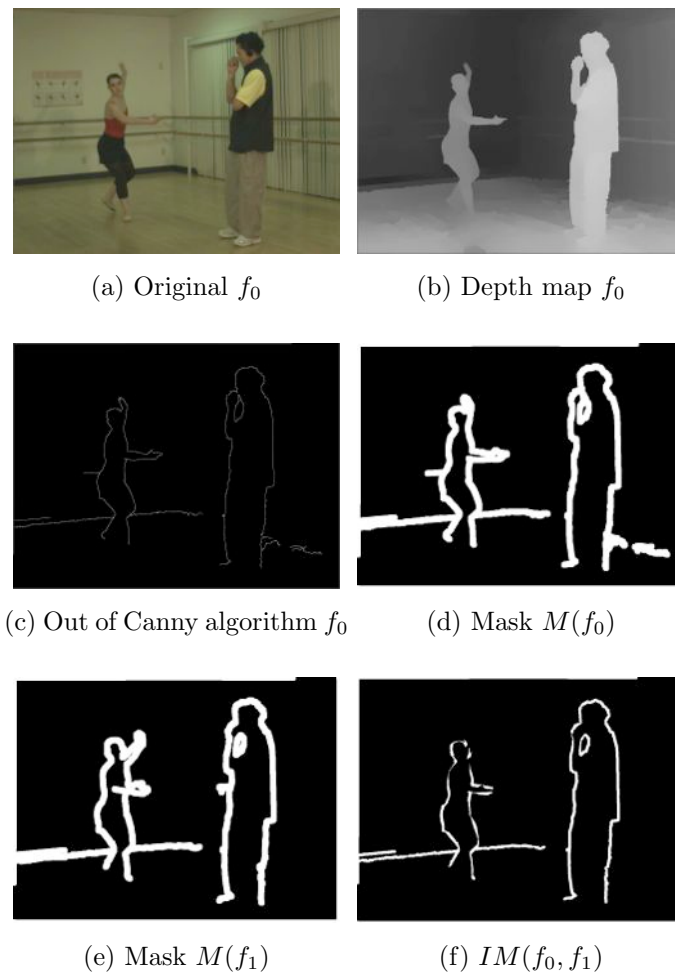


Figure 5.10: Calculation the depth relevance region of the *Ballet* sequence.

The Figure 5.10 shows the result of the application of depth relevance regions identification method for the *Ballet* sequence. The Figure 5.10a and 5.10b show the texture and depth maps of the first frame  $f_0$  of the sequence, respectively. The Figure 5.10c presents Canny's algorithm output of depth map, presented in Figure 5.10b. Figures 5.10d and 5.10e show the binary masks of frames  $f_0$  and  $f_1$ , respectively. Finally, the intersection mask  $IM(f_0, f_1)$  is presented in Figure 5.10f.

Once the regions of higher depth relevancy are identified, the DSM measure is computed based on the pixels of the two frames  $f_k$  and  $f_n$  and located inside the intersection mask  $IM(f_k, f_n)$ . The DSM is computed according to

$$DSM(f_k, f_n) = 1 - \frac{1}{2} \left[ \frac{MSE(f_k(x, y), f_n(x, y))}{A} \right] \quad (5.19)$$

$$x, y \in IM(f_k, f_n)$$

where  $A$  is normalisation factor given by the maximum MSE obtained in the 3D video temporal segment, i.e., video shot.

**Shot boundary detection algorithm** - the shot segmentation method employed here is derived from the algorithm proposed in the Section 5.1 with some modifications. The SBD algorithm is based on the measures of texture and depth variation along the temporal dimension. In the 3D video sequence i.e., stereo video and video-plus-depth, the mean of absolute difference  $d_{sad}(i)$  and the mean of absolute differences of luminance histogram  $d_{hist}(i)$  are used as the relevant metrics in the spatial dimension, while in the depth dimension, the absolute difference between the number of pixels  $d_{np}(i)$  in the binary masks  $M(f_i)$  is used. For each frame  $i$ , the mean of absolute difference  $d_{sad}(i)$  is defined as:

$$d_{sad}(i) = \frac{1}{|S|} \sum_{r \in S} |f_{i-1}(r) - f_i(r)| \quad (5.20)$$

where  $f_i(r)$  indicates the pixel value at spatial position  $r$  in the  $i$ th frame,  $S$  represents the set of pixel positions in the overall frame and  $|S|$  is the total number of pixels in a frame. The mean of absolute luminance histogram difference  $d_{hist}(i)$  for each frame  $i$  is defined as follow,

$$d_{hist}(i) = \frac{1}{N_b} \sum_{c=1}^{N_b} |H_{i-1}(c) - H_i(c)| \quad (5.21)$$

where  $N_b$  represents the total number of histogram bins, and  $H_i(c)$  represents the number of pixels in frame  $i$  contained in the  $c^{th}$  bin. For each frame  $i$ , the absolute difference of the number mask pixels  $d_{np}(i)$  is specified as,

$$d_{np}(i) = |\#pixels(M(f_{i-1})) - \#pixels(M(f_i))| \quad (5.22)$$

where  $M(f_i)$  is the binary mask of the frame  $i$  and  $\#pixels(\cdot)$  is the number of nonzero pixels in the mask. These three metrics are combined in a feature vector  $V(f_i)$ . Finally, the feature vector  $V(f_i)$  of each 3D frame  $i$  is used in a K-means clustering algorithm to determine 3D shot boundary frames, similarly to what is done in the SBD algorithm of Section 5.1.

**Texture-depth measure** - combines texture and depth information based on the underlying idea that texture and depth similarity perception can be merged together into a single 3D quality index, as proposed by [193] where a metric  $\tau$  is defined for two 3D frames  $f_k$  and  $f_n$  as

$$\tau(f_k, f_n) = T_{k,n}^\alpha Z_{k,n}^\beta \quad (5.23)$$

where  $T_{k,n}$  refers to an objective texture similarity metric between frames  $f_k$  and  $f_n$ ,  $Z_{k,n}$  is a measure which combines their relevant depth regions with the average depth level in the common region,  $\alpha$  and  $\beta$  are positive constants.

In this work  $T_{k,n}$  is measured by the Structural Similarity Index Measure (SSIM) of [194] and  $Z_{k,n}$  by the DSM measure defined before.

However since the texture-depth measure defined in Equation (5.23) is a similarity measure and we need a dissimilarity indicator we define a texture-depth dissimilarity measure  $\hat{\tau}$  based on  $\tau$  as follows:

$$\hat{\tau}(f_k, f_n) = 1 - \tau(f_k, f_n) \quad (5.24)$$

This quantity,  $\hat{\tau}(f_k, f_n)$  is a dissimilarity measure that can be used to evaluate 3D video quality in a full-reference mode if one of the input frames (say  $f_k$ ) is a reference and the other ( $f_n$ ) is the frame which quality one wants to compute. The measure output assumes values in the interval from zero to one, where zero represents the the highest quality and one the lowest.

**Key-frame extraction method** - this method selects a set of  $m$  key-frames from a video shot (temporal segment) with  $n$  frames. The selection of these 3D key-frames is based on minimisation of the global dissimilarity between the reconstructed and the original video shot in terms of the texture quality and depth perception. For that purpose, the key-frame extraction method based on minimum reconstruction error presented in the Section 5.2.1 is used. Where, the frame dissimilarity (distortion)  $d(f_k, f'_k)$  measure using  $\hat{\tau}$  defined in Equation (5.24).

## Experimental setup and datasets

For the experiments six sequences were used. This set of test data has a variety of different types of content, resolution, duration, number of frames, 3D video format and the number of video shots. Table 5.11 presents the relevant information about this test data.

Table 5.11: Characterisation of the test sequences used in the experiments.

ID	Sequence Name	Length (mm:ss)	Resolution	#Frames	Format	#Video Shots
s01	BMX	0:09	1920 × 1080@25fps	240	Stereo	4
s02	3DTestSeqA(Sofa,Feet,Hallway,Notebook,Bike,Car)	1:00	1920 × 1080@25fps	1500	Stereo	6
s03	3DTestSeqB(Akko,Exit,Ballroom,Vassar,Rena)	0:52	640 × 480@25fps	1300	Stereo	5
s04	3DTestSeqC(Box,Hall,Phone call,Lab,News,Poker)	1:00	1920 × 1080@25fps	1500	V+D	6
s05	3DTestSeqD(Ballet,Book,Kendo,Ballons)	0:32	1024 × 768@25fps	800	V+D	4
s06	3DTestSeqE(Dog,Champagne tower,Pantomime)	0:52	1280 × 960@25fps	1300	Stereo	3
Total		4:43	-	6640	-	28

The test sequences  $3DTesSeq_i$ ,  $i \in \{A, B, C, D, E\}$  were composed by concatenating the sequences indicated between parenthesis to generate scene cuts at the joining points. The ground-truth data of the 3D video transitions were obtained manually, as it is described in the Section 5.1.

The positive constants  $\alpha$  and  $\beta$  in Equation 5.23, used to weigh differently the two components of the texture-depth measure were set to 1, but other optimised values could be used for better matching with user perception of similarity.

## Results and analysis

In this section the performance of the proposed method was evaluated. For a quantitative evaluation, key-frame ratio  $R(K)$ , SRD and Fidelity measures are used. The texture-depth measure defined in Equation (5.24) is used to compute the distances between the frames  $d(\cdot)$  of the  $Fm$  measure and distortion function  $d(\cdot)$  of SRD measure.

**Key-frame ratio** - in this experiment 18 summaries from the six sequences are used, with different number of key-frames  $\{m = 2, 3, 4\}$ . The Table 5.12 shows the key-frame ratio  $R(K)$  of the all test sequences for the three  $m$ . The  $3DTestSeqE$  sequence has the lower  $R(K)$  values for each  $m$ , which is a desirable value for any summarisation method.

Table 5.12: Key-frame ratio.

ID	Sequence Name	#Key-frames		
		2	3	4
s01	BMX	3.33E-02	5.00E-02	6.67E-02
s02	3DTestSeqA	8.00E-03	1.20E-02	1.60E-02
s03	3DTestSeqD	7.69E-03	1.15E-02	1.54E-02
s04	3DTestSeqC	8.00E-03	1.20E-02	1.60E-02
s05	3DTestSeqD	1.00E-02	1.50E-02	2.00E-02
s06	3DTestSeqE	4.62E-03	6.92E-03	9.23E-03



**SRD performance** - the performance of the proposed method was evaluated by comparing the 3D key-frames summaries (proposed) against summaries generated by UnS and Clu methods, which are used as reference. The same number of key-frames is used in these three methods, i.e., proposed (Pro), UnS and Clu. In the case of UnS, the selection of key-frames is based on a constant temporal distance between frames. In the case of the Clu method, key-frame selection is based on the method proposed by [114] for 2D video. In the Clu method, the color histogram algorithm is applied only to Hue component of the HSV color space. Next, the K-means clustering algorithm is applied to cluster similar frame based on hue-colour histogram. Subsequently the frame which is closest to the cluster centroid measured by Euclidean distance was selected as a key-frame for each cluster. The number of clusters and key-frames are fixed *a priori*.

The SRD of these 18 summaries were computed and the results are shown in Table 5.15, where (max), (min) and (sd), are the maximum, minimum, and standard deviation of SRD for all test sequences. The results of Table 5.15 show that when it is increased the  $m$  the SRD values decrease for all methods. This is an expected behaviour since when  $m$  increases, the original and reconstructed shots become more similar as interpolation of missing frames benefits from the higher-rate sampling that occurs during the summarisation procedure.

To compare the performance of the proposed method with the UnS and Clu methods, it is also useful to express the results as a relative improvement measures  $\Delta_{SRD}$  and  $\Delta_{Fm}$  for the SRD and  $Fm$  measure. Tables 5.13 and 5.14 show the relative improvement of the proposed method in comparison with the UnS and Clu methods. Two relative measures are used:  $\Delta_{SRD}$  and  $\Delta_{Fm}$ . These are expressed as percentages and computed as follows,

$$\begin{aligned}\Delta_{SRD} &= \frac{SRD_{\theta} - SRD_P}{SRD_{\theta}} \times 100 \\ \Delta_{Fm} &= \frac{Fm_P - Fm_{\theta}}{Fm_{\theta}} \times 100 \quad \theta \in \{UnS, Clu\}\end{aligned}\tag{5.25}$$

where  $SRD_P$  and  $Fm_P$  are the values of SRD and  $Fm$  for the proposed method, respectively. The SRD and  $Fm$  values used in this comparison are the average values shown in the Tables 5.15 and 5.16, respectively.

From the results shown in Table 5.13, one can observe that the proposed method is always better than the UnS and Clu methods. When the proposed method is compared with UnS, the  $\Delta_{SRD}$  range lies in the interval [6.96%, 33.52%] and the best result is obtained for summaries with 3 key-frames. When the comparison is done between proposed and

Clu methods, the  $\Delta_{SRD}$  ranges from 11.07% to 32.51% and the best result is obtained for summaries with 4 key-frames.

Table 5.13: 3D summarisation -  $\Delta_{SRD}$  comparison: proposed *vs* UnS and Clu methods.

#key-frames	2	3	4	
Metric [%]	$\Delta_{SRD}$	$\Delta_{SRD}$	$\Delta_{SRD}$	Average
Proposed <i>vs</i> UnS	6.96-29.31	9.66-33.52	8.30-31.91	19.95
Proposed <i>vs</i> Clu	11.37-28.97	12.29-30.87	11.07-32.51	21.18

**Fidelity evaluation** - in the second experiment, the performance evaluation of the proposed method in comparison with UnS and Clu by quantifying the Fidelity measure is also executed. The same conditions of the first experiment were used in this trial. Table 5.16 shows the absolute values of the Fidelity measure obtained in the experiment. Contrary to SRD measure, when it is increased  $m$ , Fidelity values of the all methods in analysis increase.

Table 5.14: 3D summarisation -  $\Delta_{Fm}$  comparison: proposed *vs* UnS and Clu methods.

#Key-frames	2	3	4	
Metric [%]	$\Delta_{Fm}$	$\Delta_{Fm}$	$\Delta_{Fm}$	Average
Proposed <i>vs</i> UnS	7.86-9.51	9.31-13.19	5.64-20.70	11.03
Proposed <i>vs</i> Clu	8.14-30.20	5.61-9.24	0.57-9.11	10.48

From the results shown in Table 5.14, one can observe that the proposed method is always better than UnS and Clu methods. When the proposed method is compared with UnS, the  $\Delta_{Fm}$  range lies in the interval [7.86%, 19.02%] and the best result is obtained for summaries with 4 key-frames. The average results are quite similar to UnS when compared proposed with the Clu method. The overall results show that the frames selected by the proposed method effectively allow better reconstruction of the original shot than UnS and Clu methods.

Finally from the results, one can conclude that the key-frames extracted with the proposed method constitute an accurate representation of original 3D video shots since the SRD and  $Fm$  values of proposed method are consistently better than UnS and Clu. The results also show that the texture-depth measure can be used in objective evaluation of the performance of 3D video summarisation.

Table 5.15: SRD measure results.

Method	#Key-frames: 2									#Key-frames: 3									#Key-frames: 4											
	Pro			UnS			Clu			Pro			UnS			Clu			Pro			UnS			Clu					
	SRD			SRD			SRD			SRD			SRD			SRD			SRD			SRD			SRD					
Metric	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd
s01	0.67	0.43	0.12	0.76	0.60	0.08	0.76	0.45	0.15	0.63	0.38	0.12	0.69	0.46	0.10	0.71	0.49	0.10	0.59	0.35	0.12	0.66	0.47	0.09	0.66	0.49	0.09			
s05	0.27	0.13	0.05	0.34	0.16	0.07	0.37	0.25	0.05	0.25	0.13	0.05	0.36	0.23	0.05	0.42	0.23	0.07	0.23	0.13	0.04	0.29	0.21	0.03	0.38	0.17	0.08			
s06	0.45	0.26	0.09	0.48	0.36	0.06	0.57	0.33	0.11	0.43	0.24	0.09	0.50	0.43	0.04	0.53	0.32	0.10	0.42	0.23	0.09	0.48	0.34	0.07	0.52	0.38	0.06			
s07	0.65	0.19	0.17	0.67	0.35	0.14	0.67	0.39	0.11	0.61	0.18	0.16	0.63	0.34	0.13	0.60	0.44	0.06	0.57	0.17	0.15	0.59	0.32	0.12	0.57	0.39	0.07			
s08	0.61	0.32	0.13	0.62	0.44	0.09	0.63	0.44	0.09	0.58	0.32	0.11	0.61	0.43	0.08	0.64	0.33	0.13	0.56	0.30	0.11	0.59	0.40	0.08	0.59	0.35	0.11			
s09	0.72	0.28	0.23	0.76	0.38	0.20	0.81	0.41	0.20	0.71	0.28	0.22	0.75	0.41	0.18	0.76	0.38	0.20	0.69	0.28	0.22	0.72	0.42	0.17	0.73	0.40	0.19			
Average	0.56	0.27	0.13	0.60	0.38	0.11	0.63	0.38	0.12	0.53	0.25	0.12	0.59	0.38	0.10	0.61	0.37	0.11	0.51	0.25	0.12	0.56	0.36	0.09	0.57	0.36	0.10			

Table 5.16: Fidelity measure results.

Method	#Key-frames: 2									#Key-frames: 3									#Key-frames: 4											
	Pro			UnS			Clu			Pro			UnS			Clu			Pro			UnS			Clu					
	Fm			Fm			Fm			Fm			Fm			Fm			Fm			Fm			Fm					
Metric	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd	max	min	sd
s01	0.33	0.11	0.11	0.33	0.13	0.10	0.27	0.10	0.07	0.33	0.14	0.08	0.33	0.15	0.08	0.30	0.14	0.07	0.39	0.19	0.09	0.39	0.16	0.11	0.29	0.17	0.05			
s05	0.56	0.29	0.10	0.56	0.34	0.07	0.60	0.34	0.11	0.70	0.31	0.13	0.61	0.42	0.09	0.63	0.30	0.12	0.72	0.30	0.16	0.69	0.34	0.14	0.69	0.38	0.13			
s06	0.55	0.33	0.10	0.44	0.30	0.07	0.44	0.25	0.09	0.56	0.33	0.11	0.41	0.29	0.05	0.52	0.33	0.08	0.57	0.33	0.11	0.55	0.30	0.12	0.52	0.33	0.09			
s07	0.58	0.14	0.15	0.57	0.15	0.16	0.41	0.14	0.11	0.57	0.20	0.13	0.57	0.07	0.17	0.43	0.16	0.10	0.58	0.29	0.14	0.57	0.15	0.14	0.57	0.26	0.13			
s08	0.56	0.21	0.15	0.45	0.15	0.14	0.39	0.20	0.09	0.56	0.22	0.14	0.45	0.21	0.10	0.56	0.20	0.16	0.56	0.26	0.12	0.48	0.19	0.13	0.48	0.26	0.09			
s09	0.66	0.17	0.28	0.62	0.10	0.29	0.38	0.13	0.14	0.66	0.19	0.25	0.62	0.11	0.28	0.66	0.16	0.29	0.68	0.22	0.25	0.63	0.18	0.26	0.66	0.18	0.25			
Average	0.54	0.21	0.15	0.49	0.19	0.14	0.42	0.19	0.10	0.56	0.23	0.14	0.50	0.21	0.13	0.52	0.22	0.14	0.58	0.26	0.15	0.55	0.22	0.15	0.53	0.26	0.12			

### 5.2.4 3D and 2D key-frame extraction driven by aggregated saliency maps

This section proposes a generic framework for automatic key-frame extraction, either from 2D or 3D video sequences, using aggregated visual saliency maps to compute a perceptually meaningful distortion metric, which includes visual attention estimates in the process of key-frame selection. For each video segment (shot), a corresponding set of key-frames is chosen via dynamic programming by minimising the dissimilarity between the original video shot and the shot reconstructed from the set of key-frames. The aggregated saliency map integrates different maps of visually relevant features, allowing to discriminate between similar frames, in pixel-wise sense, but considering and distinguishing regions of different visual relevance for the human observer. In the case of 3D video, the aggregated saliency map is validated by comparison with ground-truth data from a publicly available database of fixation density maps.

In general, the main novel aspects of the proposed methods are: (i) the key-frame selection process is driven by an aggregated saliency map, computed from various feature maps, which in turn correspond to different visual attention models; (ii) a method for computing aggregated saliency maps in 3D video is proposed and validated using fixation density maps, obtained from ground-truth eye-tracking data; (iii) 3D video content is processed within the same framework as 2D video, by including a depth feature map into the aggregated saliency; (iv) objective evaluation metrics are used to test the performance of the proposed method; (v) better performance than UnS and Attention Curve (AtC) methods.

#### Proposed framework

The generic framework for key-frame extraction from classic 2D video (only one view: V1), stereo video (two views: V1 and V2) or video-plus-depth (one view: V1 and Depth) is shown in Figure 5.11. The original 2D or 3D video sequence is split into temporal shots, based on either temporal-only or depth-temporal feature clustering, respectively. The 3D video shot boundary detection algorithm previously proposed in Section 5.1 is used in this work with small changes. This algorithm combines different visual features and uses K-means clustering. Nevertheless, as other key-frame extraction methods proposed in this chapter, the proposed framework is not dependent on any particular shot boundary detection algorithm.

Recent works have concluded that aggregation of saliency maps is beneficial for improving the performance of eye-fixation prediction methods, though just increasing the number of maps does not necessarily lead to better results [195]. Following these leads, saliency map is computed by aggregating the information from two or three saliency feature maps, depending on whether 2D or 3D video is being processed, respectively. These saliency feature maps are computed from spatial, temporal and depth information and then combined into a single saliency map. This map is then further processed through a centre-bias weighting function to model the human tendency to look at the centre of an image [196]. Note however, that the centre-bias hypothesis does not hold systematically, as recently discussed in [197]. The method used to compute saliency feature maps is based on the method proposed in Section 3.1.1. As explained in the next section the resulting saliency map is then used in the process of selecting the key-frames.

**Saliency-driven distortion** - in the key-frame extraction process, a distinctive element of this work is the new distortion metric defined to drive the maximisation of similarity (by distortion minimisation) between the extracted key-frames and the corresponding video shot. In the proposed framework, such distortion metric comprises not only information about frame difference, but also the visual relevance of different image regions as estimated by the aggregated saliency map. A saliency-driven distortion metric is thus defined on a frame-basis, which integrates two types of information from each single pixel: (i) the luminance difference between frames under comparison; and (ii) the difference of visual importance between co-located pixels in each frame, as given by the corresponding aggregated saliency maps.

The relevant characteristic of this distortion metric is its ability to distinguish between frames that are pixel-wise similar but with different regions of visual interest. This allows extending the concept of frame dissimilarity (or distortion) beyond the absolute difference of their pixels, by considering that similar frames with different visual saliency maps are in fact perceived as different by human observers. For instance, in 3D video two frames might have very similar luminance and colour signals (i.e., similar texture), but quite different depth, with resulting differences in the degree and foci of attention. Therefore, from a user point of view these two images are different despite the similarity of their texture. A similar analysis can be done for 2D video taking into account accumulated motion, for instance.

The proposed saliency-driven distortion function is defined as the sum of two terms, where

the saliency difference between two frames is used as a pixel-wise weighting factor for the frame similarity measure and at the same time the frame dissimilarity is added as an independent term on its own. This function is given by Equation (5.26), for any two frames  $f_k, f'_k$ , with spatial resolution  $h \times w$ ,

$$d(f_k, f'_k) = \frac{1}{hw} \sum_{j=0}^{h-1} \sum_{i=0}^{w-1} \left[ dp_k(i, j) + (1 - dp_k(i, j)) ds_k(i, j) \right] \quad (5.26)$$

with

$$dp_k(i, j) = \frac{|f_k(i, j) - f'_k(i, j)|}{255} \quad (5.27)$$

and

$$ds_k(i, j) = \left| S_G^{f_k}(i, j) - S_G^{f'_k}(i, j) \right| \quad (5.28)$$

where  $S_G^{f_k}(i, j)$  is the saliency value for each pixel located at  $(i, j)$  in frame  $f_k$ .

Using the above definition given by Equation 5.26, the higher the pixel-wise dissimilarity between frames, the lower is the importance of the visual saliency. For instance, in the extreme case where the pixel difference is maximum (i.e.,  $dp_k(i, j) = 1$ ), the dissimilarity is also maximum, regardless of the saliency weight  $ds_k(i, j)$ . This means that, if the visual contents of two frames are completely different from each other, the corresponding saliency difference is not a meaningful dissimilarity measure and the frame difference on its own is enough to measure the perceptual dissimilarity.

On the other hand, the difference in visual saliency becomes more relevant as the similarity between frames increases. For instance, in the extreme case of any two pixels with equal values (i.e.,  $dp_k(i, j) = 0$ ), the dissimilarity becomes dependent on the saliency factor  $ds_k(i, j)$ . This means that, if two frames are equal and their saliency maps are also equal, then  $d(f_k, f'_k) = 0$ , i.e., minimum dissimilarity. However, if two frames are equal and their saliencies are different, then this means that they are perceived as different frames with the level of dissimilarity given by the saliency value  $ds_k(i, j)$  in Equation 5.26.

This distortion metric is used as input to an optimisation algorithm based on dynamic programming, which finds the indices of the best key-frames to be selected from the original shot. The algorithm, described in Section 5.2.1, chooses the key-frames such that the overall distortion between the original shot and the shot reconstructed from the key-frames is as small as possible.

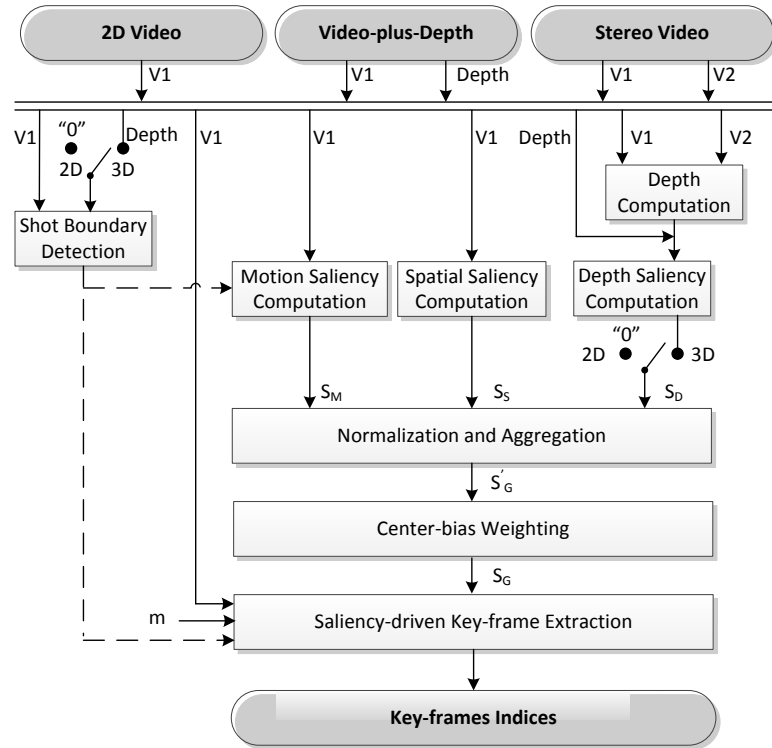


Figure 5.11: Framework for 2D/3D key-frame extraction.

**Computation of saliency feature maps** - as mentioned above, the aggregated saliency map results from fusion of three different feature maps, which are estimates of visual relevancy measured in the spatial (texture), motion and depth domains. The spatial and motion feature maps,  $S_S$  and  $S_M$  respectively, are computed from a single view, while the depth feature map  $S_D$  is derived from the depth maps. Both stereoscopic video and video-plus-depth can be used to obtain the depth related saliency information, with the former requiring prior depth computation from the stereoscopic views. Then, normalisation and fusion of these feature maps generates the intermediate map  $S'_G$  which is further weighted with a centre-bias weighting function to generate the final aggregated visual saliency map  $S_G$ . More details about the method to compute the saliency feature maps, see Section 3.1.1 of the Thesis, since method used here is the one described in that section.

**Key-frame extraction method** - involves selecting the set of  $m$  key-frames which best represent a temporal shot of  $n$  frames. The number of key-frames  $m$  to be selected can be given as a user-defined parameter or computed according to some predefined criteria. The method used for identification of the representative key-frames is based on the minimisation of the saliency-driven dissimilarity between frames of the original shot

and the corresponding ones reconstructed from the set of key-frames. For that purpose, the key-frame extraction method based on minimum reconstruction error presented in the Section 5.2.1 is used. Where, the frame dissimilarity (distortion)  $d(f_k, f'_k)$  that quantifies the difference between frame  $f_k$  of  $F$  and its corresponding frame  $f'_k$  from  $F'$  measure using  $d(f_k, f'_k)$  defined in Equation (5.26). The dynamic programming algorithm is used to solve the key-frame ratio-dissimilarity optimisation problem. For more details of this method, see the Section 5.2.1 of the this document.

### Experimental setup and datasets

For the experiments with 2D video, six sequences of the Open Video Project (the first six of Table 5.17) [198] and four sequences from [62] were used. These video databases were chosen because they provide user-defined summaries to be used as ground-truth for performance evaluation. A similar evaluation methodology was used in [114]. Additionally, this set of test data has a variety of different types of content, such as documentary, education, historical, lecture and ephemeral. For the experiments with 3D video, seven test sequences were used namely, *Boxers*, *Hall*, *Phone call*, *Laboratory*, *News report* from the NAMA3DS1 database [199], *Poker* sequence from the European FP7 Project MUSCADE [200] and *Poznan hall 2* from the Poznan multiview video database [201]. Table 5.17 summarizes the relevant information about these sequences.

Table 5.17: Details of the test sequences.

ID	2D Sequence Name	#Frames	Resolution
s1	The Great Web of Water-seg.01	3279	352 × 240@30fps
s2	A New Horizon-seg.02	1797	352 × 240@30fps
s3	Oceanfloor Legacy-seg.01	1740	352 × 240@30fps
s4	Drift Ice as a Geologic Agent-seg.10	1407	352 × 240@30fps
s5	Hurricane Force - A Coastal Perspective-seg.03	2310	352 × 240@30fps
s6	The Future of Energy Gases-seg.09	1884	352 × 240@30fps
s7	Canada Day	765	352 × 240@30fps
s8	Dragon Boat	702	352 × 240@30fps
s9	Walk with the Dragon	461	352 × 240@30fps
s10	Nitobmov	792	352 × 240@30fps
3D Sequence Name			
s11	Boxers	250	1920 × 1080@25fps
s12	Hall	250	1920 × 1080@25fps
s13	Phone call	250	1920 × 1080@25fps
s14	Laboratory	250	1920 × 1080@25fps
s15	New report	250	1920 × 1080@25fps
s16	Poker	250	1920 × 1080@25fps
s17	Poznan hall2	200	1920 × 1088@25fps



For the computation of visual saliency maps, the weights  $w_s$ ,  $w_m$  and  $w_d$  of Equation (3.6) of Section 3.1.1 were assigned constant values, following the underlying idea that motion features are more relevant than others [202]. It was found empirically that the results are not critically dependent on small variations of these weights and after experimentation the values  $w_s = w_m = 0.5$  for 2D video and  $w_s = w_d = 0.25$ ,  $w_m = 0.5$  for 3D video were chosen to be used in the remaining experiments. Note that these weights are regarded as system parameters and different criteria can be used to choose their values, either fixed or adaptive. To compute the motion saliency map  $S_M$  (using block-matching), square blocks with size equal to 16 pixels were used. The spatial saliency map  $S_S$  was computed using Hou’s model [29].

## Results and analysis

This section describes the experiments carried out to evaluate the performance of the proposed methods, including saliency map aggregation and saliency-driven 2D/3D key-frame extraction.

**Aggregated saliency maps for 3D video** - the performance of the method proposed for computing the aggregated saliency map for 3D video was evaluated by comparing the computed saliency maps against the ground-truth fixation density maps obtained from the eye-tracking data from a dataset recently made available [1]. In this dataset the fixation density maps are obtained by post-processing the gaze points recorded from the left and right eyes, which are combined into a map of single gaze points using the stereo disparity. Then, similarly to the method proposed in [203], a gaussian filter is applied, followed by normalisation between 0 and 255. All 3D video sequences presented in Table 5.11 were used to validate the aggregated saliency maps. The spatial saliency computation methods from Itti [22], Hou [29] and Bruce [30] are also used as references for comparison. Note that, although these are 2D image-based methods, they provide known references for performance comparison in studies like [2] where they were compared with methods to compute saliency maps of 3D images.

Table 5.18 shows the PLCC and KLD values obtained with the use of nine different methods used to compute saliency maps for 3D video including the proposed aggregation method. The results of both Wang’s method to compute stereoscopic 3D video visual saliency [2] and our proposed method are presented in Table 5.18 separated into three

different cases, each one corresponding to frame-level spatial saliency maps  $S_S$  obtained using Itti [22], Hou [29] and Bruce's [30] methods.

The results in Table 5.18 show that the method proposed achieves a maximum average PLCC of 0.436 and minimum average KLD of 0.598 when using Hou's method, i.e., *Proposed(Hou)*. This method has similar PLCC and KLD values for *Boxers*, *Hall*, *Phone call* and *News report* test sequences. For sequences *Laboratory*, *Poker* and *Poznan hall2* the PLCC achieved by *Proposed(Bruce)* is the best while for KLD, the method *Wang(Hou)* gives the best results. Overall, the average values of PLCC and KLD shown in Table 5.18 show that for the three cases under study the proposed method achieves better results than the competing methods.

From these results conclude that the 3D video saliency maps obtained with the proposed method show marginally better PLCC and KLD performance than the others. In all likelihood this improvement is due to the aggregation of the three saliency feature maps, demonstrating that it is important to fuse different visual cues when modeling and computing saliency, as the visual saliency map computed by the proposed method is closer to the ground-truth fixation density maps obtained from the gazing preferences of human viewers.

Table 5.18: Aggregated saliency maps for 3D video: performance comparison.

Method	Itti[22]		Hou[29]		Bruce[30]		Wang(Itti) [2]		Wang(Hou) [2]		Wang(Bruce) [2]		Proposed(Itti)		Proposed(Hou)		Proposed(Bruce)	
Metric	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD	PLCC	KLD
Boxers	0.185	1.449	0.247	0.566	0.282	2.547	0.266	1.177	0.307	0.607	0.315	2.031	0.602	0.828	0.654	0.713	0.582	1.327
Hall	0.123	1.732	0.357	0.639	0.237	3.427	0.197	1.435	0.361	0.638	0.239	2.626	0.333	1.257	0.451	0.596	0.381	1.687
Phone call	0.297	1.207	0.526	0.704	0.386	2.039	0.307	1.133	0.527	1.097	0.431	1.419	0.547	0.502	0.584	0.409	0.516	0.798
Laboratory	0.090	1.205	0.081	0.766	0.324	2.934	0.166	1.445	0.118	0.700	0.326	2.107	0.264	1.188	0.262	0.802	0.385	1.497
News report	0.413	0.876	0.449	0.391	0.424	2.925	0.414	1.051	0.457	0.399	0.401	2.518	0.404	1.040	0.492	0.375	0.403	1.975
Poker	0.153	0.644	0.351	0.535	0.396	1.900	0.167	0.532	0.325	0.303	0.375	1.474	0.167	1.052	0.331	0.876	0.387	1.451
Poznan hall2	0.082	0.327	0.204	0.388	0.307	2.486	0.090	0.321	0.214	0.498	0.316	2.075	0.182	1.152	0.276	0.415	0.351	1.806
Average	0.192	1.063	0.316	0.570	0.337	2.608	0.230	1.013	0.330	0.606	0.343	2.036	0.357	1.003	<b>0.436</b>	<b>0.598</b>	0.429	1.506

Table 5.19: 2D key-frame extraction: proposed *vs* non-saliency based methods.

Method	OV[115]		DT[133]		STIMO [134]		VSUMM [114]		Proposed	
Metric	$CUS_A$	$CUS_E$	$CUS_A$	$CUS_E$	$CUS_A$	$CUS_E$	$CUS_A$	$CUS_E$	$CUS_A$	$CUS_E$
The Great web water-seg.01	0.55	0.73	0.20	0.37	0.83	1.02	0.75	0.96	0.72	0.84
A New Horizon-seg.02	0.33	0.09	0.15	0.18	0.38	0.12	0.61	0.14	0.84	0.07
Oceanfloor Legacy-seg.01	0.23	0.36	0.36	0.53	0.61	1.17	0.75	0.43	0.81	0.60
Drift Ice as a Geologic Agent-seg.10	0.96	0.09	0.96	0.09	0.84	0.21	0.75	0.30	0.75	0.09
Hurricane Force-seg.03	0.59	0.25	0.64	0.20	0.36	0.61	0.70	0.14	0.81	0.45
The Future of Energy Gases-seg.09	0.03	3.47	0.55	0.70	0.67	0.83	0.42	0.58	0.53	0.72
Average	0.45	0.83	0.48	<b>0.35</b>	0.62	0.66	0.66	0.43	<b>0.74</b>	0.46

Table 5.20: 2D key-frame extraction: proposed *vs* saliency based methods.

Method	Lai and Yi[62]		Peng and Xiao-Lin[85]		Proposed	
	$CUS_A$	$CUS_E$	$CUS_A$	$CUS_E$	$CUS_A$	$CUS_E$
The Great web water-seg.01	0.71	0.85	0.71	0.69	0.72	0.84
Hurricane Force-seg.03	0.78	0.75	0.58	0.53	0.81	0.45
Canada Day	-	-	0.75	0.25	1.00	0.00
Dragon Boat	-	-	0.75	0.25	0.63	0.13
Walk with the Dragon	-	-	0.75	0.25	0.75	0.00
Nitobmov	-	-	0.75	0.25	0.75	0.00
Average	0.75	0.80	0.75	0.43	<b>0.78</b>	<b>0.24</b>

**2D/3D key-frame extraction** - this section presents the experimental results and performance evaluation of the key-frame selection process based on the proposed framework. To make the presentation and analysis clearer, the 2D and 3D cases are presented and analysed in separate subsections. Three quality measures have been used to evaluate the quality of key-frame summaries: CUS, SRD and  $Fm$ . CUS is a quantitative measure of similarity of two summaries proposed in [114] and expressed by two values, the accuracy rate  $CUS_A$  and the error rate  $CUS_E$ . More details about CUS measure see Section 2.4.4. SRD measures the degree to which a set of key-frames is a good representation of the original shot by computing an average distance or distortion between the original shot and a shot reconstructed from the key-frames. Here, the distance or distortion function  $d(\cdot)$  of SRD measure is the saliency-driven distortion defined in Eq.(5.26). Finally,  $Fm$  is based on the Semi-Hausdorff distance between the key-frame set and the original shot. This is computed as the maximum of the minimal distances between the set of key-frames and each frame of the original shot [136]. The saliency-driven distortion defined in the Eq.(5.26) is also used to compute the distances between the frames  $d(\cdot)$  of the  $Fm$  measure. All details of these quality measures have been reported in Section 2.4.4.

**2D video** - in this section the key-frame summaries computed by the proposed framework are compared with six other methods: OV [115], DT [133], STIMO [134], VSUMM [114], Lai and Yi [62] and Peng and Xiao-Lin [85]. The last two methods also use visual saliency.

User-defined summaries provided in [114] were used as references for comparison. This ground-truth data was manually extracted by five different users after watching a down-sampled sequence of frames taken from the original video. For a fair comparison between the different key-frame extraction methods and these reference user summaries, each 2D video sequence was also down-sampled following the same procedure as in [114].

Six sequences were chosen to compare our proposed method with some prominent non-visual attention methods. Table 5.19 presents the accuracy rate  $CUS_A$  and error rate  $CUS_E$  obtained for the six test sequences, using four methods (none using saliency) to extract the key-frames, plus the proposed one. Note that the better key-frames summaries are those that present low  $CUS_E$  and high  $CUS_A$ . The average values indicate that the proposed framework achieves the highest  $CUS_A$  (0.74) whilst DT [133] achieves the lowest  $CUS_E$  (0.35). In the DT method, the lower  $CUS_E$  value is justified by the smaller number of selected key-frames, in comparison with the user summaries. Consequently, the DT key-frames present a low  $CUS_E$  at the cost of a low  $CUS_A$ .




























































As shown in Table 5.19, the  $CUS_A$  achieved by the proposed framework is higher than the  $CUS_A$  achieved by VSUMM [114]. For  $CUS_E$  some of the values obtained by method VSUMM are better than those of the method proposed but the average values are very similar. This is due to the fact that the proposed method selects more key-frames than other methods as at least one key-frame per shot is always selected. Consequently, more summary key-frames do not have a match in the reference summary leading to larger  $n_{no-match}$  values and to higher  $CUS_E$  values. However, considering both  $CUS_A$  and  $CUS_E$  the proposed framework provides better results than the competing methods. In addition, the proposed framework also ensures the chronological order of the key-frames, which is not guaranteed by the methods based on clustering [114, 133, 134].

Table 5.20 presents the results obtained for the six 2D test sequences using two different visual attention methods as references to compare the extracted key-frame summaries with the proposed one. The key-frame summaries produced by the Peng and Xiao-Lin [85] and Lai and Yi [62] were provided in [63, 64, 85]. On average, the table shows that the proposed framework yields better  $CUS_A$  (0.78) and  $CUS_E$  (0.24) than the other methods. From the results presented in the Tables 5.19 and 5.20 one concludes that the quality and accuracy of the key-frame summaries produced by the proposed framework are compare favorably to those produced by the reference methods.

Finally, in Table 5.21 the key-frames extracted by all methods as well as the ground-truth key-frames selected by #User5 [114] for the documentary *Hurricane Force - A Coastal Perspective*, segment 03 are shown. The ground-truth summary, used as reference for comparison with the other summaries, comprises 8 key-frames. As shown in Table 5.21, some important frames are missing in summaries OV (i.e., frames 3,7 and 8) and DT (i.e., frames 4,5 and 8). STIMO also misses important frames, including the one showing the first person (i.e., frames 1,2,3,4,5 and 8). In the case of VSUMM, even though it

generates a fairly good summary, i.e., average  $CUS_A = 0.66$  and  $CUS_E = 0.43$  (see Table 5.19), only 6 frames were selected and some important ones are also missing (i.e., 1,4 and 5). The results of the Peng and Xiao-Lin [85] and the proposed framework are visually comparable but the method of Lai and Yi [62] selects more three frames than the ground-truth. Overall, one can observe that, in comparison to the other methods, the key-frames extracted by the proposed framework are closer to the ground-truth selected by users.

Table 5.21: Key-frames extracted from sequence video *Hurricane Force - A Coastal Perspective*, segment 03.

Method	Generated Key-frames								
	1	2	3	4	5	6	7	8	
<b>Ground Truth #User5</b>									
OV[115]									
DT[133]									
STIMO[134]									
VSUMM[114]									
Peng and Xiao-Lin [85]									
Lai and Yi [62]									
<b>Proposed</b>									

**3D video** - for 3D video, the performance of the generic key-frame extraction framework is evaluated by comparing the output of the proposed framework with that of a UnS and AtC. The two measures introduced before, SRD and  $Fm$ , are used to quantify the summaries quality. The same number of key-frames  $m$  is used in the comparison of the three methods. In the case of UnS, the key-frames extracted are temporally equidistant, i.e., are uniformly distributed along the original video shot. For the AtC method, the saliency maps of each frame are used to generate an attention curve which is then used to select the key-frames that will make up the summary. The number of key-frames is user-defined and those frames having the highest attention values in each shot are selected as key-frames. A similar approach was used in [59, 62–64, 85]. In this experiment 21 summaries from seven sequences are used, with  $\{m = 2, 3, 4\}$  i.e., key-frame ratio  $\frac{2}{250}, \frac{3}{250}$  and  $\frac{4}{250}$ .

Table 5.23 shows the results of SRD and  $Fm$  metrics for all test sequences. From the results one can observe that increasing  $m$  leads to a decrease in SRD for all methods.

The Fidelity measured by  $Fm$  also increases with  $m$ . To compare the performance of the proposed method with the UnS and AtC methods, it is also useful to express the results as a relative improvement measures  $\Delta_{SRD}$  and  $\Delta_{Fm}$  for the SRD and  $Fm$  measure. Table 5.22 shows the relative improvement of the proposed method in comparison with the UnS and AtC methods. Two relative measures are used:  $\Delta_{SRD}$  and  $\Delta_{Fm}$ . These are expressed as percentages and computed as follows,

$$\begin{aligned}\Delta_{SRD} &= \frac{SRD_{\theta} - SRD_P}{SRD_{\theta}} \times 100 \\ \Delta_{Fm} &= \frac{Fm_P - Fm_{\theta}}{Fm_{\theta}} \times 100 \quad \theta \in \{UnS, AtC\}\end{aligned}\tag{5.29}$$

where  $SRD_P$  and  $Fm_P$  are the values of SRD and  $Fm$  for the proposed method, respectively. The SRD and  $Fm$  values used in this comparison are the average values shown in the Table 5.23.

From the results shown in Table 5.22, one can observe that the proposed method is always better than the UnS and AtC methods. When the proposed method is compared with UnS, the  $\Delta_{SRD}$  range lies in the interval [11.34%, 14.13%] and the best result is obtained for summaries with 4 key-frames. In the case of  $\Delta_{Fm}$ , the proposed method is also better than UnS and the best result is 1.76% for summaries with 2 key-frames. The results are even better when compared with the AtC method, since the key-frames extracted are, in some cases, located near each other. In this case, the  $\Delta_{SRD}$  ranges from 14.17% to 26.01% and the best result is obtained for summaries with 4 key-frames. For  $\Delta_{Fm}$ , the best result is 3.23%, obtained for summaries with 3 key-frames. The overall results show that the key-frames selected by the proposed method effectively allow better reconstruction of the original shot than UnS and AtC methods. Therefore, the key-frames selected by the proposed method constitute a more accurate representation of the original shot, since the SRD and  $Fm$  values are consistently better than UnS and AtC.

Table 5.22: 3D key-frame extraction: proposed *vs* UnS and AtC methods.

Metric [%]	#Key-frames:2		#Key-frames: 3		#Key-frames: 4	
	$\Delta_{SRD}$	$\Delta_{Fm}$	$\Delta_{SRD}$	$\Delta_{Fm}$	$\Delta_{SRD}$	$\Delta_{Fm}$
Proposed <i>{vs}</i> UnS	11.51	1.76	11.34	0.74	14.13	1.13
Proposed <i>{vs}</i> AtC	14.17	1.75	21.03	3.23	26.01	2.63

Table 5.23: Performance of the proposed key-frame extraction method for 3D video.

Method	#Key-frames: 2						#Key-frames: 3						#Key-frames: 4					
	Proposed		AtC		UnS		Proposed		AtC		UnS		Proposed		AtC		UnS	
Metric	SRD	Fm	SRD	Fm	SRD	Fm	SRD	Fm	SRD	Fm	SRD	Fm	SRD	Fm	SRD	Fm	SRD	Fm
Boxers	0.085	0.858	0.089	0.825	0.096	0.831	0.080	0.866	0.088	0.826	0.095	0.859	0.078	0.874	0.087	0.843	0.085	0.855
Hall	0.051	0.906	0.078	0.878	0.071	0.902	0.043	0.915	0.074	0.883	0.062	0.912	0.037	0.920	0.073	0.887	0.055	0.919
Phone call	0.074	0.890	0.102	0.890	0.084	0.868	0.070	0.897	0.082	0.890	0.078	0.884	0.067	0.907	0.072	0.901	0.080	0.889
Laboratory	0.107	0.833	0.111	0.808	0.128	0.856	0.090	0.864	0.106	0.808	0.099	0.859	0.079	0.876	0.102	0.857	0.100	0.859
News report	0.054	0.934	0.063	0.932	0.069	0.923	0.050	0.937	0.059	0.933	0.059	0.934	0.049	0.944	0.058	0.934	0.058	0.934
Poker	0.173	0.782	0.208	0.781	0.185	0.751	0.147	0.810	0.207	0.781	0.155	0.797	0.131	0.814	0.206	0.782	0.145	0.813
Poznan hall2	0.177	0.776	0.189	0.762	0.182	0.746	0.151	0.786	0.184	0.762	0.166	0.784	0.137	0.797	0.182	0.769	0.150	0.794
Average	0.103	0.854	0.120	0.839	0.116	0.839	0.090	0.868	0.114	0.841	0.102	0.861	<b>0.083</b>	<b>0.876</b>	0.112	0.853	0.096	0.866



The comparative results provide empirical evidence that better performance is obtained from fusion of different visual cues when modeling and computing saliency. It was shown that aggregation of texture, motion and depth produces more accurate saliency maps, i.e., in better agreement with corresponding saliency data obtained from actual human viewers. Overall this framework extends existing ones beyond the spatial and temporal dimensions by considering aggregated saliency maps and saliency-driven distortion in the process of optimal selection of 2D/3D key-frames.

## 5.3 Conclusions

In this chapter, a framework for represent in compact way 2D and 3D video was introduced. This framework is composed with two main steps. Firstly, video sequence is segmented in temporal segments which are compound by frames with similar content. This temporal segmentation is based in several features and it is also adapted to different video formats, i.e., 2D and 3D video. Second, a method which selects the most representative frames of each temporal segments, based on a specific distortion metric, is too presented.

In case of 3D video, the temporal segmentation algorithm detects 3D video shot boundaries based on joint depth-temporal criterion. The experimental results show that the proposed algorithm is capable of accurately detecting 3D shot boundaries, exhibiting good performance measured as recall and precision rate. The results also showed that 3DSB detection algorithm is independent of the 3D video content and format because its decision process is not based on thresholds neither training data and it can be used stereoscopic and video-plus-depth video. Some adaptation, as removal depth information from temporal segmentation criteria, were made in the SBD algorithm, to be used for 2D video. Due to the flexibility of proposed algorithm two metrics were used to compute the depth information and the results show good performance for these two metrics.

Selection of the most representative frames of each video shot is based on minimum reconstruction error for that, three methods key-frame extraction methods, based on MSE, PCA, perceptually relevant depth regions and aggregated saliency maps are presented. Experimental results and performance comparison, based on multiple metrics, demonstrate that these methods outperforms other methods proposed in the literature for similar purposes of extracting key-frames from 2D or 3D video.



# Flexible video coding based on spatial and temporal scalability

---

The large diversity and varying characteristics of multimedia content used in contemporary communication services and applications requires efficient and flexible management. This flexibility can be achieved by exploiting the fact that not all information contained in video sequences is equally important and relevant. Representation methods based on this tenet can not only provide high flexibility in the access to the most relevant content but also achieve better coding efficiency. In this chapter, novel flexible video coding methods are proposed based on spatial and temporal scalability, integrating the methods for video summarisation and ROI detection described in earlier chapters.

Two flexible ROI (video) coding methods are proposed, the  $QP_{51}$  and *Set-to-Zero* methods. Both methods are characterized by the use of a low resolution constant quality base layer complemented by a higher resolution layer, which represents pixels of ROIs with higher spatial fidelity. In both methods the higher layer bits are spent mainly in encoding the ROI, minimizing the number of bits spent in the background non-ROI regions.

Additionally, a temporal flexible video coding scheme for video summaries is proposed which reduces the problems derived from the lack of flexibility of the traditional video coding schemes on what concerns the definition of temporal prediction structures. Most video coding schemes use regular GOP structures which do not accommodate efficiently variable temporal rates and flexible prediction structures needed for independent coding of video summaries. The proposed solution encodes video summaries using dynamic GOP sizes coupled with a temporal scalability scheme. The video summary is encoded as temporal base layer and the remainder non-summary frames are encoded using higher temporal layer(s). The experimental results show the effectiveness of the novel flexible video coding methods.

Part of the experiments and results presented in this chapter were published in C6, C7 and C10.

## 6.1 ROI coding with spatial scalability

This section describes two methods for coding ROIs, based on H.264/MPEG-4 AVC and spatial scalability. Their rate-distortion performance and complexities are presented and compared to a reference coder. In both methods the base layer is kept unchanged and provides a lower resolution video encoded with constant quality, agnostic to the presence of any ROI. In either case there is no need to encode contour information because the ROI is implicitly defined in the upper spatial resolution layer in a transparent way by using different encoding parameters for the ROIs and their complementary regions. In Section 2.5 it is shown that spatial scalability can be used to encode efficiently specific regions of an image sequence with differentiated spatial resolution and quality according to the regions importances.

The underlying idea to achieve efficient coding of the ROI in the higher resolution layer is to minimise the number of bits spent in the background region of the higher resolution images. One of the methods proposed in this work is based on the use in the higher resolution layers of coarse quantisation for the background non-ROI regions and finer quantisation of the ROIs. Two methods are proposed:  $QP_{51}$  and *Set-to-Zero*. In  $QP_{51}$ , the Macro-Blocks (MB) of the background region, i.e., outside the ROIs, are coded with the maximum quantisation value allowed by H.264/MPEG-4 SVC ( $Q_p=51$ ) in order to maximise the number of null coefficients. In *Set-to-Zero*, the transform coefficients of the MBs outside the ROI are set to zero regardless their value. Note that in this case quantisation and coding is in fact not performed for these MBs. In both methods, the ROIs are defined by a mask, providing a ROI map which is used by the encoder to identify the ROIs MBs. The identity of the MBs belonging to the ROIs is not encoded into the video stream and is used only at encode time.

### 6.1.1 $QP_{51}$

The functional implementation of this method is depicted in Figure 6.1. During encoding of each MB of the high resolution layer, the QP value is switched between 51 and the QP value selected for the current MB, depending on whether the MB is located outside the ROIs or inside a ROI, respectively. Therefore, the quality of ROI MBs is much higher than that of the MBs outside the ROI and consequently most of the bits used in the high resolution layer are assigned to the ROI. Note that in the high resolution layer the only

useful information that needs to be coded is the ROI itself, because the lower quality and resolution of the background region provided by the base layer should be enough for the envisaged application. The encoding of the base layer lower resolution video is oblivious to the presence and position of ROIs.

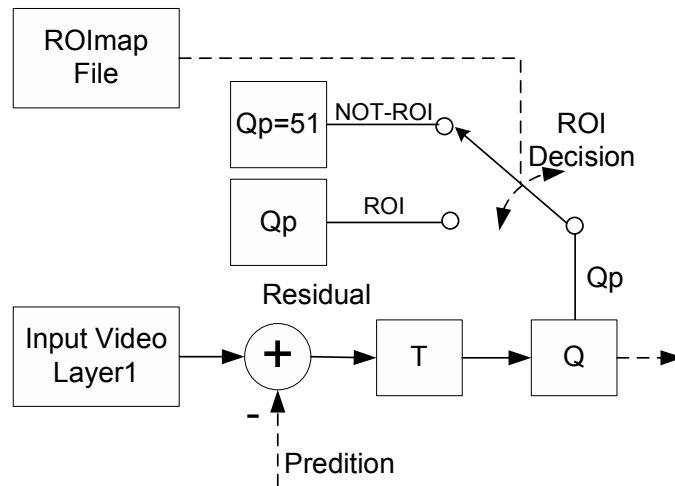


Figure 6.1:  $Qp_{51}$  functional diagram.

### 6.1.2 Set-to-Zero

The objective of the *Set-to-Zero* method is the same as the previous one: to spend less bits encoding the MBs outside the ROIs than those inside the ROIs and so to increase the quality of ROI representation in the higher resolution layer. In this method, the transform coefficients of residual blocks are set to zero for those MB outside the ROIs. Since H.264/MPEG-4 SVC uses the syntax element Coded Block Pattern (CBP) to indicate which  $8 \times 8$  blocks of a MB contains non-zero coefficients, in the *Set-to-Zero* method the encoder sets CBP to zero to signal an all-zero MB. Figure 6.2 shows *Set-to-Zero* functional diagram.

### 6.1.3 Results and analysis

The performances of the two methods described in the previous section were evaluated in regard to R-D and computational complexity, measured as the processing time per frame, and compared with straightforward coding without ROI.

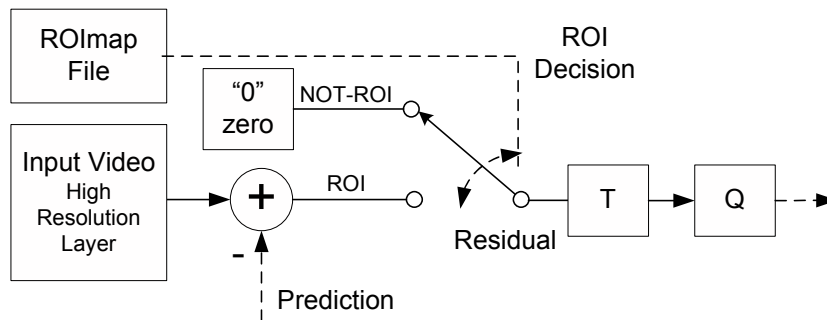


Figure 6.2: *Set-to-Zero* functional diagram.

Separate experiments were carried out for intra and inter coding modes. The proposed methods were implemented using the JVT reference software, version 8.9, as a basis framework. The test sequence *Mobile* was used in the experiments with resolution QCIF@30fps for base layer and CIF@30fps for enhancement layer. Two ROIs (ROI1, ROI2) with different sizes were used. Figure 6.3 shows ROI1 on the left image, as the set of pixels inside the red rectangle, and ROI2 on the right image also outlined in red. In this study, the ROIs were defined manually, but other more sophisticated methods such as the method of Section 4.1.4 could be used as the ROI coding approaches proposed are not dependent on the method used to generate the ROI. In the experiments the following settings were used:

- Intra test - two spatial layers (QCIF and CIF) at 30fps; 100 frames; NumberReferenceFrames 1; FastSearch; Loop Filter on. The coding parameters were as follows: for the base layer: CABAC; Basic QP 35; FRExt no; for higher layer: CABAC; InterLayerPred on; FRExt on.
- Inter test - two spatial layers (QCIF and CIF) at 30fps; 100 frames; NumberReferenceFrames 1; FastSearch; Loop Filter on; MaxDelay 1200; GOPsize 16; IntraPeriod 16. The coding parameters were as follows: for the base layer: CABAC; Basic QP 35; FRExt no; for higher layer: CABAC; InterLayerPred on; FRExt on.

The simulations were performed on a PC with a 2.4 GHz processor and 1.0 GB of RAM memory.

The bitrate shown in Figures 6.4 and 6.5 is the sum of the bitrates of the base layer and of the higher layer. The bitrate values shown were obtained using different QPs in the higher layer while the QP of base layer is constant (QP=35). The ROI PSNR (i.e., the PSNR computed for the pixels within the ROI of the higher layer) is shown in Figures 6.4

and 6.5 for intra and inter coding, respectively. For reference, the two proposed methods results are compared with results from an experiment where the higher layer is entirely coded using the same QP without distinguishing between ROI and background. These reference results are labelled as *SVC-without\_ROI*.

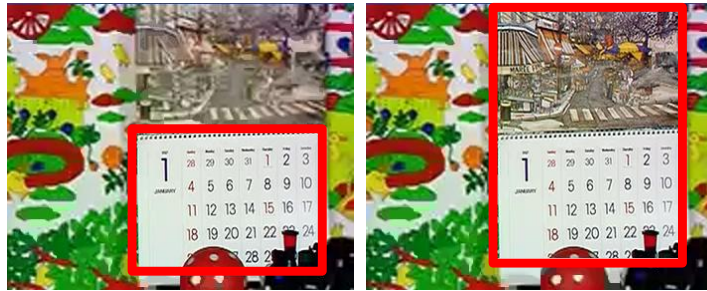


Figure 6.3: ROI1 (left) and ROI2 (right) of the *Mobile* sequence outlined in red.

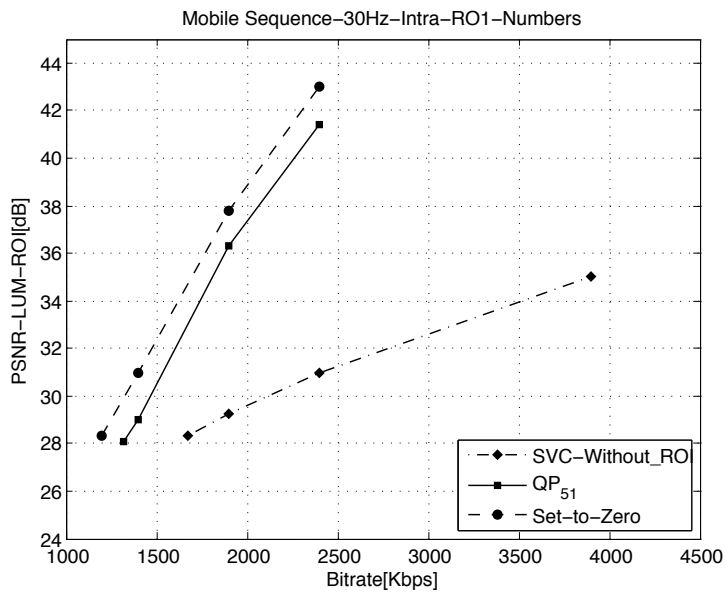
### Intra coding

The R-D performance of the intra case is shown in Figure 6.4. The *Set-to-Zero* method is compared with  $QP_{51}$  and *SVC-without\_ROI*. The coding complexity is shown in Table 6.1 for both ROIs. From Figures 6.4a and 6.4b it is clear that the efficiency of the *Set-to-Zero* method is consistently better for both ROIs in the intra case. In ROI1 this method produces a PSNR about 2dB higher than the  $QP_{51}$  method. As one can see in the figures, the overall quality gain of the proposed methods is much higher when compared to *SVC-without\_ROI*. For the lower bitrates in ROI1, the *Set-to-Zero* method produces a PSNR about 6.5dB higher than *SVC-without\_ROI* and at higher bitrates the gain is about 13dB. For the ROI2 the gains of *Set-to-Zero* are smaller than in the case of ROI1. About 0.4dB - 0.5dB higher than and 2.5dB - 7.5dB higher than *SVC-without\_ROI* for low and high bitrates, respectively. To the same PSNR, both the  $QP_{51}$  method and *SVC-without\_ROI* produce more bits than *Set-to-Zero* for coding ROI1 and ROI2.

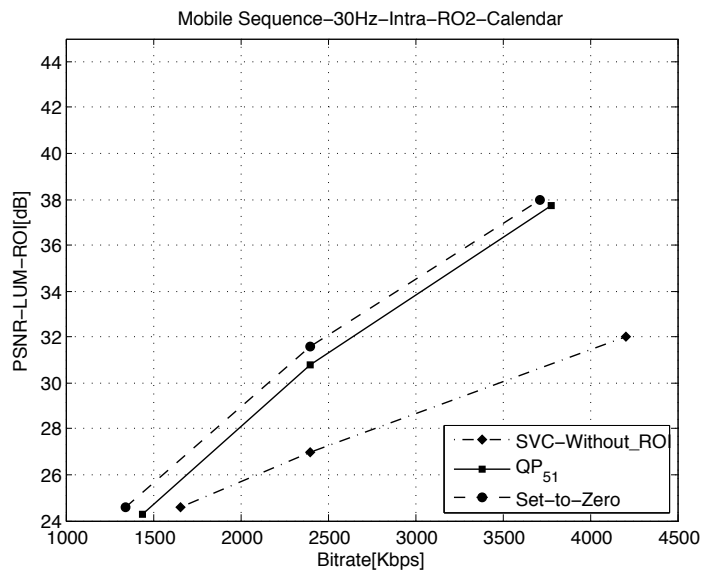
Table 6.1 shows the processing time of the two methods and the reference *SVC-without\_ROI*. From the table, it is easy to conclude that the coding complexity of the *Set-to-Zero* method is lower than that of the other two ( $QP_{51}$  and *SVC-without\_ROI*) for both ROIs. For ROI1, the processing time is reduced by 12% to 30% with *Set-to-Zero* compared to *SVC-without\_ROI* and by 7% compared to the  $QP_{51}$  method. In the case of ROI2, the processing time of *Set-to-Zero* method is reduced 9% to 22% compared to *SVC-without\_ROI* and 5% compared to  $QP_{51}$ . The lower complexity of the *Set-to-Zero* is due mainly to the quantisation not being computed for the MBs outside the ROI with a significant reduction in the number of computations.

Table 6.1: Processing time of the intra coding mode.

	QP	<i>Set-to-Zero</i> [ms/frame]	$QP_{51}$ [ms/frame]	<i>SVC-without_ROI</i> [ms/frame]
ROI1	25	182.53	195.83	262.26
	35	174.37	187.35	225.24
	45	167.54	179.58	192.12
ROI2	25	204.97	217.07	262.26
	35	189.10	198.85	225.24
	45	174.37	182.66	192.12
Average		<b>182.15</b>	193.56	226.54



(a) ROI1-Numbers



(b) ROI2-Calendar

Figure 6.4: R-D performance for the intra coding case.



### Inter coding

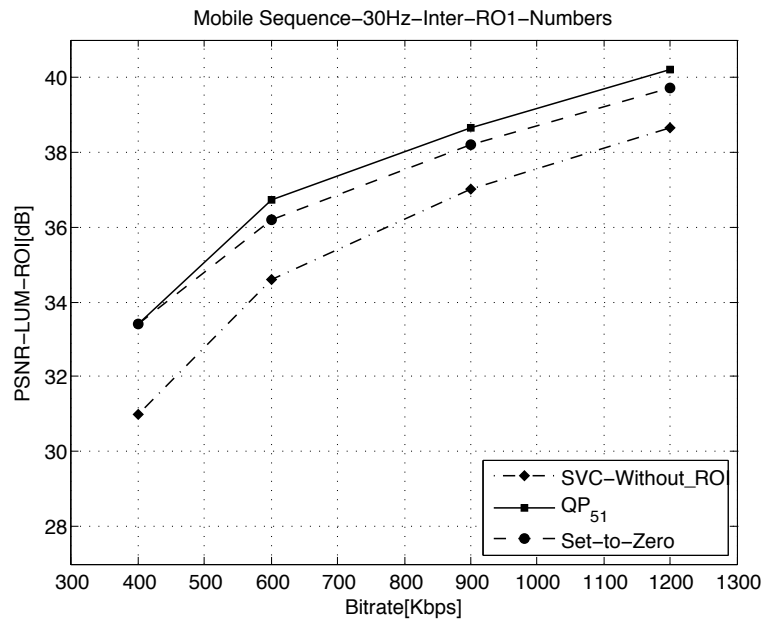
The performance of inter coding is shown in Figure 6.5. In this case, the efficiency of *Set-to-Zero* is closer to that of  $QP_{51}$ . For the encoding of ROI1 the gains of both proposed methods are practically the same for low bitrates, while for higher bitrates the  $QP_{51}$  method produces gains of about 0.8dB and 1.2dB compared with *Set-to-Zero* and *SVC-without\_ROI*, respectively. In the case of ROI2, the *Set-to-Zero* yields better results relatively to the other methods and it is about 0.4dB better than  $QP_{51}$  and nearly 2.6dB better than *SVC-without\_ROI*.

Table 6.2, shows that the processing time depends on the ROI dimension, the QP and the coding methods used. In this case, the processing times are larger than those in the intra case. Also as in the intra case, the *Set-to-Zero* method is less complex than the other methods as can be seen from the values presented in Table 6.2 for all experiments.

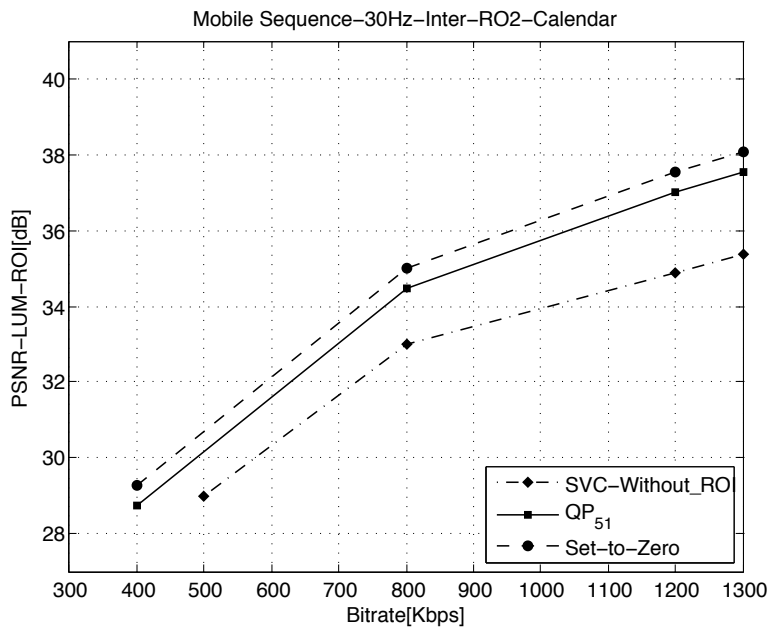
Table 6.2: Processing time of the inter coding mode.

	QP	<i>Set-to-Zero</i> [ms/frame]	$QP_{51}$ [ms/frame]	<i>SVC-without_ROI</i> [ms/frame]
ROI1	25	168.72	169.05	170.31
	35	168.72	169.03	169.41
	45	168.72	168.87	169.40
ROI2	25	168.74	169.05	170.31
	35	168.73	169.04	169.41
	45	168.71	169.03	169.40
Average		<b>168.72</b>	169.01	169.71

The performance of the proposed ROI coding methods shows that spatially scalable ROI encoding can be performed with very good results by using selective coding for each region in the higher resolution layer. The results obtained also show that the *Set-to-Zero* method is computationally less complex than  $QP_{51}$ , which makes it a good candidate for software-based implementations. By keeping the coded stream fully compatible with the H.264/MPEG-4 SVC standard, the proposed methods are suitable for a wide range of applications where only specific regions of a video sequence are needed at higher spatial resolution e.g., remote surveillance, video conferencing, medical applications and others.



(a) ROI1-Numbers



(b) ROI2-Calendar

Figure 6.5: R-D performance for the inter coding case.

## 6.2 Video summary coding with temporal scalability

A method to efficiently encode an arbitrary video summary with temporal scalability and dynamic GOP structures is proposed in this section. The video summary is coded as the base layer and the remainder frames are coded in the upper layers of a SVC bitstream. The video frames of the summary are identified and the respective temporal indices are input to the proposed scalable encoder which computes dynamically the GOP size based on a coding efficiency criterion which takes into account the MSE between the frames of the summary.

### 6.2.1 Dynamic GOP size selection

In general, the GOP structure used in scalable video coding is fixed and regular over time in order to provide a hierarchical coding structure [11]. In this type of GOP structure the number and type of frames (either  $P$  or  $B$ ) are predefined as encoder configuration parameters. The  $I$  frames determine the GOP boundaries. All type of frames are allowed in the temporal base layer, i.e.,  $I$ ,  $B$  and  $P$  frames, while  $P$  and  $B$  frames are coded only in the upper layers following a regular structure over the whole sequence. Note, however, that such regular GOP structure is not mandatory to be compliant with the SVC extension to the standard. Since a video summary does not have a regular frame rate, rather than using a fixed GOP structure, it is better to use GOPs of variable size according to the temporal distribution of the video summary frames.

In order to match the variable temporal distribution of the video summary frames to a GOP structure, the total number of  $B$  and  $P$  frames within a GOP must also be variable. Therefore in a dynamic GOP structure, the number of  $B$  frames between  $I$  or  $P$  frames and the number of  $P$  frames between  $I$  frames is variable, depending on the video summary frames.

The essential motivations for using dynamic GOP structures is to not only achieve temporal scalability but also higher coding efficiency. Grouping similar frames in the same GOP will help to improve bit saving as dissimilar frames can not be efficiently coded using temporal prediction and so should be coded as  $I$  frame at the GOP boundaries. If the most dissimilar frames within a limited set of summary frames (i.e., the maximum allowed GOP size) are selected for the GOP boundaries, then good coding efficiency is

expected because all the remaining frames are the most similar ones, which will favour efficient temporal prediction.

The proposed algorithm searches for the best GOP boundaries in the video summary frames in order to achieve high coding efficiency, both in the base layer (i.e., the video summary) and in the whole sequence (i.e., all layers). To find the GOP boundaries, the summary frame distortion  $D_c$  was used, defined in Equation (6.1), where  $d(f_c, f_j)$  is the MSE, between the candidate GOP boundary frame  $f_c$  and all possible video summary frames  $f_j$  within a maximum distance of 32 frames. In the expression,  $c$  represents the index of the candidate summary frame and  $A$  is the set of ordered summary frame indices,  $A = \{l_0, l_1, \dots, l_{m-1}\}$ , such that  $l_0 < l_1, \dots, < l_{m-1}$ . Note that  $l_0, l_1, \dots, l_{m-1}$  are defined in the original frame sequence, thus  $A$  does not comprise an arithmetic progression.

$$D_c = \sum_{\substack{c-32 \leq j \leq c+32 \\ j \in A}} d(f_c, f_j)_{MSE}; \quad c = l_0, l_1, \dots, l_{m-1} \quad (6.1)$$

$D_c$  is computed for all summary frames and the best upper boundary frame index for GOP  $n$ ,  $l_n^*$  is given by,

$$l_n^* = \arg \max_c (D_c) \text{ where } l_{n-1}^* < c \leq l_{n-1}^* + 31 \quad (6.2)$$

$l_{n-1}^*$  is the lower boundary frame index of GOP  $n$ , which is also the best upper boundary frame index of GOP  $n - 1$ . In the first GOP, the lower boundary frame is  $l_0$ . In this work, the maximum allowed GOP size is 32, since this size provides enough variation headroom for the GOP size and good coding efficiency [11]. The summary is determined before coding, though some isolated frames can be inserted in the summary during coding to respect the maximum GOP size restriction. In the cases where consecutive summary frames have a temporal distance higher than 32 (e.g., key-frames or video skims summaries), the algorithm forces the GOP size to take the value of 32 effectively promoting a non-summary frame to a summary frame.

### 6.2.2 Prediction structure in temporal scalable coding

Figure 6.6 shows the prediction structure which results from the dynamic GOP size allocation using the method described above. As shown in the figure, the video summary frames (thicker lines in the figure) are coded in the temporal base layer while the upper layer

contains the remaining frames of the sequence. Therefore, the full temporal resolution is obtained when both temporal layers are decoded. Since the reference frames of the video summary are all coded in the temporal layer 0, the summary layer can be extracted and independently decoded from the whole coded stream. The R-D performance of coding a GOP depends not only on the coding order but also on the employed reference frames. However, in general, as the temporal interval between each frame and its reference gets shorter, temporally predictive coding becomes more efficient. Therefore, among the available frames in the Decoded Picture Buffer (DPB), we choose the nearest frames of the current one to be used as its forward and backward references. In this work, all frames between the GOP boundary frames are coded as B, though they can also be coded as  $P$  type.

In H.264/MPEG-4 AVC with scalable extension, the proposed scheme changes the default order of reference pictures in List0 for the  $P$  slices or in List0/List1 for  $B$  slices. For a correct decoding, the reference frames of the video summary must be signaled to the decoder using Reference Picture List Re-ordering [204].

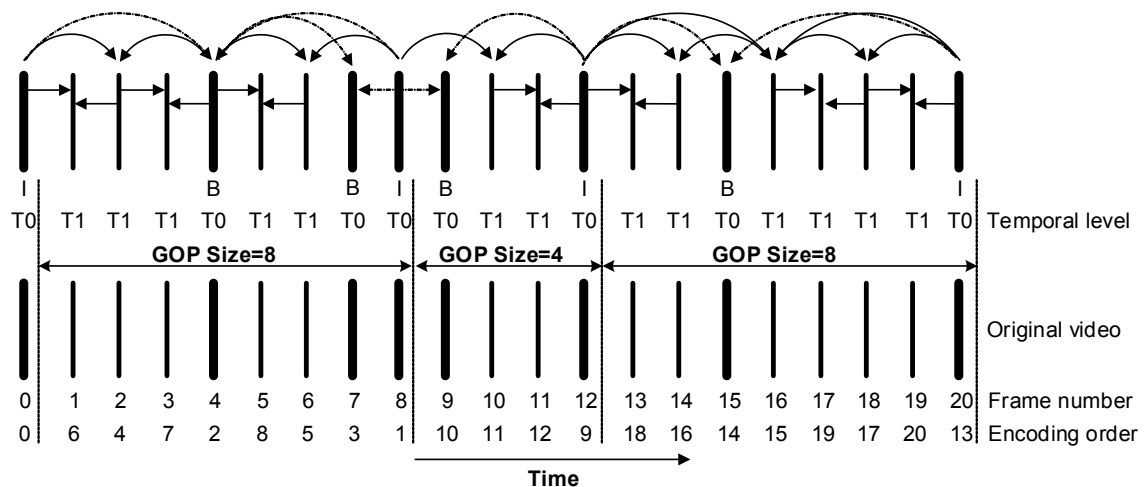


Figure 6.6: Example of a prediction structure resulting from dynamic GOP allocation.

### 6.2.3 Results and analysis

The proposed method was implemented in the Joint Scalable Video Model (JSVM) 8.9 reference software, and the test sequences *Soccer* and *Foreman*, QCIF@30Hz were used in the experiments. The main coding parameters used in the simulations were: *NumberReferenceFrames* 2; *FastSearch*; *Loop Filter* on; *CABAC*; *FRExt* no; *MaxDelay* 1200.

The R-D operational points used to draw the R-D graphs were obtained using the set of  $QP : \{25, 30, 35, 40, 45\}$ . For each sequence, three different video summaries were generated, using the algorithm proposed in Section 5.2.2, with temporal rates  $R(S)$  of 25%, 12% and 6%. For comparison, a temporally subsampled version of each sequence, acting as a reference summary, was also coded as the base layer of a temporally scalable bitstream at the same  $R(S)$ , i.e., with the same total number of frames, using the fixed GOP structure of SVC. This reference summary also represents the whole sequence by a subset of frames with the same size and provides the same functionality of being independently decodable.

### Efficiency of video summary coding

The coding efficiency observed when coding the video summaries with the proposed base layer dynamic GOP structure was compared with that of the standard SVC regular GOP structure used to encode a uniform distribution summary with the same temporal rate  $R(S)$ . This is actually a different summary because it is comprised of regularly spaced frames that are not always coincident with the frames of the other summary. However this is still a fair comparison because the same number of frames are used in both cases to represent the whole sequence.

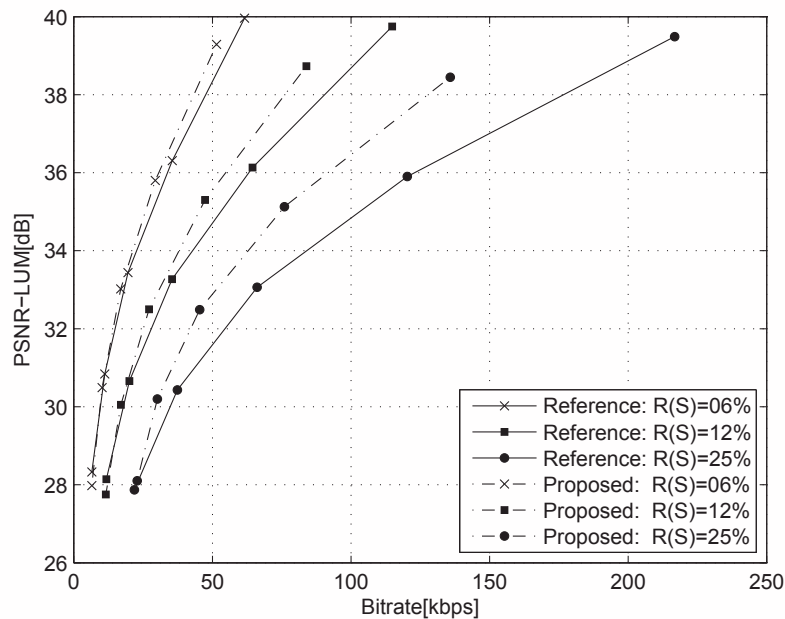


Figure 6.7: R-D performance of the temporal base layer (T0) of the *Soccer* sequence.

Figure 6.7 and 6.8 show that better efficiency is obtained using the proposed method, compared with that of SVC with fixed GOP size. The difference between the two methods is small for the two sequences with temporal rate  $R(S) = 6\%$ , but in the case of temporal rate  $R(S) = 12\%$ , the PSNR gain for *Foreman* is about 2 to 3dB, while for *Soccer* this gain ranges from 0 to 1.5dB. In the experiments with a temporal rate of 25%, the gains are higher than the  $R(S) = 12\%$ . In these tests, the PSNR gain for *Foreman* is about 3 to 5dB, while for the *Soccer* this gain ranges from 0 to 2.5dB. This is because the proposed method finds the best GOP size within each interval of 32 frames and B frames find good predictions, whereas in the case of the reference summary with regular GOP structure the GOP size can not be optimized and the long temporal distance between reference frames makes *B* frames not useful in temporal layer 0.

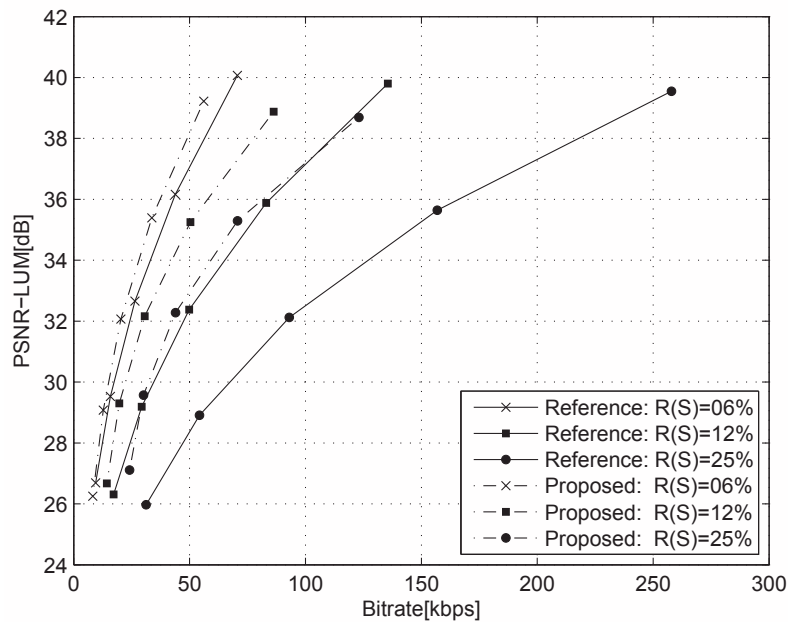


Figure 6.8: R-D performance of the temporal base layer (T0) of the *Foreman* sequence.

### Full temporal resolution coding efficiency

The overall coding efficiency was also evaluated for the full temporal rate, i.e., the whole sequence (all layers) using the proposed method to encode the base layer (with dynamic GOP) was compared with a reference using encoded a regular GOP size of 32. Figures 6.9 and 6.10 show the results for summaries of different temporal rates  $R(S)$ . The results show that for  $R(S) = 6\%$ , the coding efficiency achieved by the proposed method is virtually the same as the reference sequences while for other rates the difference is negligible.

Therefore, using the proposed method to encode summaries in the base layer does not have any noticeable impact in the overall R-D coding performance. Therefore using temporal scalability with dynamic GOP size to encode summaries achieves the sought encoding flexibility without coding performance penalties.

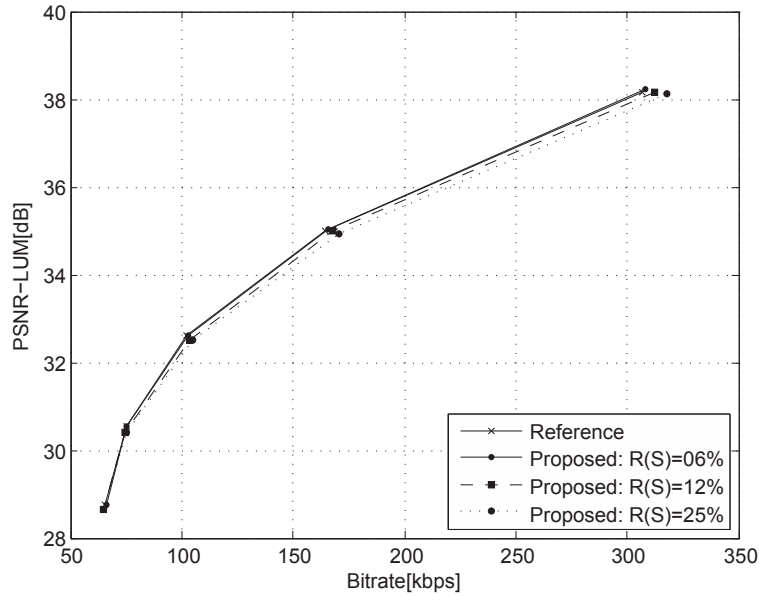


Figure 6.9: R-D of the full rate of the *Soccer* sequence.

A method to encode an arbitrary video summary using dynamic GOP structures in scalable streams was presented in this section. The scalable bitstream obtained is fully compatible with the scalable extension of the H.264/MPEG-4 AVC standard. The results show that good coding efficiency is achieved for arbitrary video summaries without compromising the quality of the whole sequence. The proposed method demonstrates that an extra level of flexibility can be achieved by embedding video summaries in scalable streams, which is of practical interest in content adaptation systems and applications.



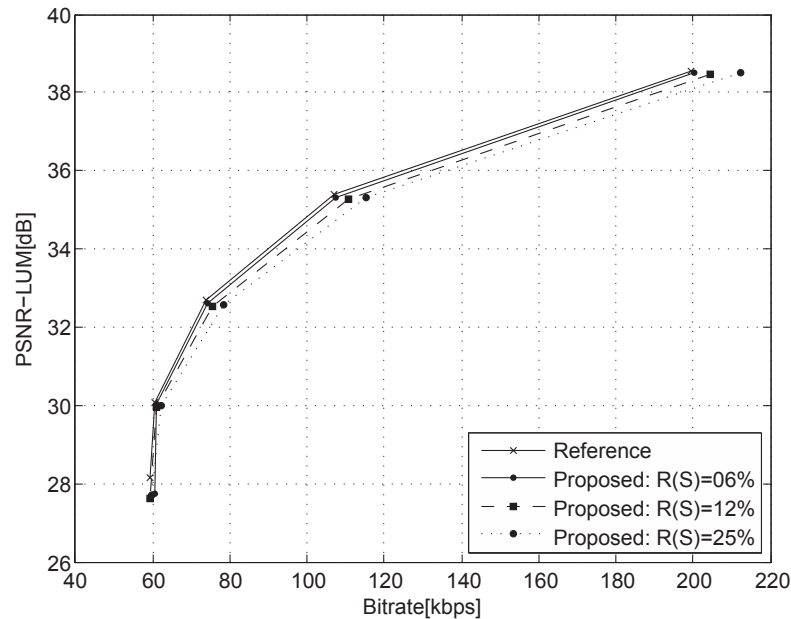


Figure 6.10: R-D of the full rate of the *Foreman* sequence.

## 6.3 Conclusions

The video coding methods introduced in this chapter seek to encode the most relevant information contained in a video sequence using scalability, in order to improve flexibility of access and user experience without hurting rate-distortion performance. New flexible video encoding methods based on the H.264/MPEG-4 SVC standard were proposed in order to generate spatially and temporally scalable bitstreams representing the output of the previously proposed methods for video retargeting and summarisation or non-specific ROI identification methods.

Two different approaches were used for encoding ROIs in a scalable framework where the most important information, i.e., the ROI was encoded with increased spatial detail in a higher resolution layer. The base layer is kept unchanged and provides lower resolution images with constant quality, without identification of the ROIs, serving as a lower quality base signal which can be complemented by decoding the higher layer information to reconstruct the pixels in the ROI at higher spatial resolution.

Additionally, a temporal flexible video summary method was proposed, based on dynamic GOP size selection. Using this scheme video summaries are encoded in the base layer of a temporally scalable stream whilst the remaining non-summary frames are encoded in the upper layers. This coding scheme allows visualisation of the video summary without

decoding the entire stream and when combined with the information from the higher layers can be used to reconstruct the full-temporal-rate video.

The experimental results have shown the effectiveness of the proposed flexible video coding methods compared to reference procedures. It is worth noting that the proposed methods are independent of the way the ROI and video summaries are obtained, which means that the present proposal can be effectively used in flexible video coding environments.

# Conclusion and future work

---

The research work developed within the scope of this Thesis contributed to advance state-of-the-art methods capable of providing additional flexibility in the representation of visual information either in raw or coded formats. The main contributions of this Thesis include both 2D and 3D video, in visual saliency computation, video retargeting, video summarisation methods and also enhanced video coding to efficiently accommodate the corresponding information and new data structures, i.e., video summary and ROI. This chapter concludes the Thesis with a summary of the main achievements, as well as some directions for future work.

## 7.1 Conclusions

The introductory chapters established the context and motivation of the Thesis, starting by the basic concepts used in representation of the visual information either in raw or coded format. A review of different state-of-the-art methods used to represent visual information with increased flexibility beyond the pixel/frame level, provided the necessary background and established the starting point of this research.

In Chapter 3, two visual saliency computation methods based on fusion of four intermediate saliency features maps (spatio-temporal, depth and face saliency) followed by a centre-bias weighting function were proposed for 3D video. The selective combination of the features maps allows the proposed methods to be applied either in 2D or 3D video. The results demonstrated that the proposed method outperforms other state-of-the-art methods. The saliency maps obtained from this work were used as input in new video retargeting and video summarisation methods developed in Chapter 4 and 5 respectively.

In Chapter 4, two spatio-temporal retargeting methods based on salient region were proposed. These methods change the resolution of original video for a specific display size. Although these proposals were developed and tested for UHD video as input, they can be

also applied to different video resolutions. A visual comparison of the outputs generated by proposed methods show the ability to preserve the relevant content of the UHD scene in comparison to others well-known methods. Furthermore, a method to improve temporal consistency through filtering for jitter removal, revealed a double benefit, both for the visual quality and for the coding efficiency. It was found that using HEVC to encode the retargeted UHD video, consistently better PSNR is obtained when temporal consistency is enforced.

In Chapter 5, a computational framework to obtain video summaries was devised. This framework is composed by two main modules. In the first module, video sequence is divided into temporal segments which are composed by frames with similar content. This video segmentation is based on depth-temporal features and can be used with different 3D video formats. In the second module, a key-frame extraction method selects the most representative frames of each temporal segments, using a dynamic programming algorithm based on rate-distortion approach that minimises the dissimilarity (distortion) between the original sequence and the one reconstructed from the key-frames. The experimental results and performance comparison, based on multiple metrics such Precision Rate, Recall Rate, SRD, Fidelity, CUS and compression ratio, demonstrate that this framework outperforms other methods presented in the current literature for similar purpose.

In Chapter 6, two methods for encoding ROI and video summary with H.264/MPEG-4 SVC encoder were devised. These methods are independent from the approach used to generate ROI and video summary and the type of summary, i.e., either *key-frames* or *video skims*. In the case of scalable coding of video summaries, the proposed method encodes the video summary in the base layer, allowing to extract a short representation of the video sequence from the coded stream. The scalable bitstream obtained by this method is fully compatible with the scalable extension of the H.264/MPEG-4 AVC standard. The results demonstrated that the proposed method can be used to encode video summaries with increased efficiency in the temporal base layer and negligible loss of R-D performance in the whole scalable sequence. Similar scalable principle was used to encode ROI. In this case, two coding approaches were employed: the  $QP_{51}$  *outside ROI* and *Set-to-Zero*. The results obtained by the two approaches are always better than the reference *SVC-without ROI*. In this comparison the R-D and processing time were the performance metrics used in these experiments.

## 7.2 Future work

The research carried out in this Thesis does not provide definite solutions for all issues addressed in the different topics. Therefore, further research is envisioned to improve current performance and provide better solutions for several aspects in related fields.

- **Visual saliency computation** - Our method combines different saliency features and fused them to get a final saliency map. A linear combination is used to fuse these saliency features. So, it would be interesting to test others combinations strategies, as it was proposed in [205], to know which combination of features gives the most robust results. Other interesting point is the inclusion of text saliency feature in our proposed method. This new approach will be designed for deaf applications where text is the most important focus.
- **Video retargeting** - The proposed method provides spatio-temporal adaptation based on visual saliency information to resize UHD video to small screen devices. The validation of the proposed method was based on visual comparison to other solutions, however, it is important to know the capability of the method to keep the visual interesting areas in the retargeted sequence. This study requires a comparison with eye-tracking data, and this is open for future work. Another issue is the objective assessment of proposed method in comparison to other approaches, since there are no well-accepted objective metrics for quality evaluation of retargeted video.
- **Temporal segmentation** - The accuracy of the SBD algorithm can be further improved, particularly new efficient methods to detect smooth transitions between shots. The shot boundary detection should be made in the temporal-depth dimension, where depth and temporal features can be used in this process. A hierarchical detection scheme could bring benefits in performance by first implementing the algorithm to detect sharp transitions and then analysing the remaining frames to detect smooth transitions.
- **Video summarisation** - Further investigation is needed to efficiently compute video skim summaries. Another point which is not addressed in video summarisation framework developed in this Thesis, is the combination of the visual features with additional information (audio features, text captions and content description). There is a lack of these type of summarisation methods (key-frame based or video

skirts) for 3D video formats such as, MVD and holoscopic video.

- **Flexible video coding** - This Thesis does not address a solution for joint encoding of ROI and video summary. It is reasonable to expect that combined solutions may perform better than using isolate methods. More combinations can be done to increase flexibility, such as salable coding of 3D video with ROI, scalable coding of 3D video summaries, scalable coding of 2D/3D video summaries and ROI.

Overall, the above issues are research topics that can be investigated to further extend the work done in this Thesis. Nevertheless, more fundamental research is required to take the flexible representation of visual information up to another level, where more human factors can be deeply embedded in the representation formats. For instance, emotional and cognitive factors are still far from being fully utilized for the benefit of representation and coding of visual information.

# Published papers

---

## A.1 Journal papers

- J1** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “A generic framework for optimal 2D/3D key-frame extraction driven by aggregated saliency maps”, *Signal Processing: Image Communication*, vol. 39, Part A, pp. 98-110, November 2015.
- J2** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “Towards key-frame extraction methods for 3D video: A review”, *EURASIP Journal on Image and Video Processing*, (submitted).

## A.2 E-letter

- E1** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “UHD video retargeting based on visual attention models and temporal consistency”, *IEEE COMSOC MMTC E-Letter*, (submitted).

## A.3 Conference papers

- C1** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “An improved method to compute visual saliency in 3D video”, Proc. CONFTELE 2015, Aveiro, Portugal, Vol. 1, pp. 1 - 4, Sep., 2015.
- C2** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “A method to compute saliency regions in 3D video based on fusion of feature maps”, IEEE International Conference on Multimedia and Expo (ICME2015), Turin, Italy, pp.1-6, Jun. 29 -Jul. 3, 2015.
- C3** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “3D Video Shot Boundary Detection Based on Clustering of Depth-Temporal Features”, International Workshop on Content-Based Multimedia Indexing - CBMI, Veszprém, Hungary, Jun., 2013.

- 
- C4** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “3D Video Key-Frame Selection based on Colour and Depth”, Conference on Telecommunications - CONFTELE, Castelo Branco, Portugal, pp.165-168 May 2013
- C5** - Pedro D. F. Correia, **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, Vitor Silva, “Optimal priority MDC video streaming for networks with path diversity”, Proc. International Conference on Telecommunications and Multimedia (TEMU), Heraklion, Greece, pp.54-59, Jul. 2012.
- C6** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “Efficient Scalable Coding of Video Summaries using Dynamic GOP Structures”, Proc EUROCON and CONFTELE 2011, Lisbon, Portugal, Vol. 1, pp. 1 - 4, Apr., 2011.
- C7** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “A Method to Encode Video Summaries Using Temporal Scalability”, Proc Workshop on Picture Coding and Image Processing - WPCIP, Nagoya, Japan, Dec., 2010.
- C8** - Luís Coelho, Luís A. Cruz, **Lino M. M. Ferreira**, Pedro A. Amado Assunção, “An Improved Sub-optimal Video Summarization Algorithm”, Proc International Symp. Electronics in Marine - ELMAR, Zadar, Croatia, Vol. 1, pp. 135 - 138, Sep., 2010.
- C9** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “Video Summary Generation and Coding Using Temporal Scalability”, Proc Conf. on Telecommunications - CONFTELE, Santa Maria da Feiria, Portugal, Vol. 1, pp. 283 - 286, May, 2009.
- C10** - **Lino M. M. Ferreira**, Pedro A. Amado Assunção, Luís A. Cruz, “H.264/SVC ROI encoding with spatial scalability”, Proc INSTICC International Conf. on Signal Processing and Multimedia Applications - SIGMAP, Porto, Portugal, Vol. 1, Jul., 2008.



## References

- [1] P. Hanhart and T. Ebrahimi, “Eyec3D: 3D video eye tracking dataset,” in *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, Sep. 2014, pp. 55–56.
- [2] Junle Wang, M.P. DaSilva, P. LeCallet, and V. Ricordel, “Computational model of stereoscopic 3D visual saliency,” *Image Processing, IEEE Transactions on*, vol. 22, no. 6, pp. 2151–2165, 2013.
- [3] Yuming Fang, Junle Wang, Jing Li, R. Pepion, and P. Le Callet, “An eye tracking database for stereoscopic video,” in *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*, Sep. 2014, pp. 51–52.
- [4] Zhu Li, G.M. Schuster, A.K. Katsaggelos, and B. Gandhi, “Rate-distortion optimal video summary generation,” *Image Processing, IEEE Transactions on*, vol. 14, no. 10, pp. 1550–1560, Oct. 2005.
- [5] Jianfeng Xu, Toshihiko Yamasaki, and Kiyoharu Aizawa, “Temporal segmentation of 3D video by histogram-based feature vectors,” *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 19, no. 6, pp. 870–881, Jun. 2009.
- [6] A. Rav-Acha, Y. Pritch, and S. Peleg, “Making a long video short: Dynamic video synopsis,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, June 2006, vol. 1, pp. 435–441.
- [7] K. Schoeffmann, D. Ahlstrom, and M.A. Hudelist, “3D interfaces to improve the performance of visual known-item search,” *Multimedia, IEEE Transactions on*, vol. 16, no. 7, pp. 1942–1951, Nov 2014.
- [8] Hwan-Jik Lee, Hyun Joon Shin, and Jung-Ju Choi, “Single image summarization of 3D animation using depth images,” *Computer Animation and Virtual Worlds*, vol. 23, no. 3-4, pp. 417–424, 2012.
- [9] Heiko Schwarz and Mathias Wien, “The scalable video coding extension of the H.264/AVC standard,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 135, 2008.
- [10] Zafar Shahid, Marc Chaumont, and William Puech, “Scalable Video Coding,” in *Effective Video Coding for Multimedia Applications*, Sudhakar Radhakrishnan, Ed., p. 18. Apr. 2011.
- [11] H. Schwarz, T. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” *IEEE Tran. on CSVT*, vol. 17, no. 9, pp. 1103–1120, 2007.

- 
- [12] ITU-R, “Parameter values for ultra-high definition television systems for production and international programme exchange,” Recommendation BT.2020-2, Oct. 2015.
- [13] ITU-R, “Parameter values for the HDTV standards for production and international programme exchange,” Recommendation BT.709-9, Jun. 2015.
- [14] ITU-R, “Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios,” Recommendation BT.601-7, Mar. 2011.
- [15] A. Smolic and P. Kauff, “Interactive 3-D video representation and coding technologies,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 98–110, Jan. 2005.
- [16] Toshiaki Fujii and Masayuki Tanimoto, “Free viewpoint TV system based on ray-space representation,” in *ITCom 2002: The Convergence of Information Technologies and Communications*. International Society for Optics and Photonics, 2002, pp. 175–189.
- [17] A. Smolic, K. Mueller, P. Merkle, P. Kauff, and T. Wiegand, “An overview of available and emerging 3D video formats and depth enhanced stereo as efficient generic solution,” in *Picture Coding Symposium, 2009. PCS 2009*, May 2009, pp. 1–4.
- [18] Christoph Fehn, “Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV,” in *Electronic Imaging 2004*. International Society for Optics and Photonics, 2004, pp. 93–104.
- [19] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, “Intermediate view interpolation based on multiview video plus depth for advanced 3D video systems,” in *2008 15th IEEE International Conference on Image Processing*, Oct 2008, pp. 2448–2451.
- [20] Kenton McHenry and Peter Bajcsy, “An overview of 3D data content, file formats and viewers,” *National Center for Supercomputing Applications*, vol. 1205, 2008.
- [21] Caroline Conti, Luís Ducla Soares, and Paulo Nunes, “3D holoscopic video representation and coding technology,” in *Novel 3D Media Technologies*, Ahmet Kondoz and Tasos Dagiuklas, Eds., pp. 71–96. Springer New York, 2015.
- [22] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [23] Anne M. Treisman and Garry Gelade, “A feature-integration theory of attention,” *Cognitive Psychology*, vol. 12, no. 1, pp. 97 – 136, 1980.
- [24] A. L. Yarbus, *Eye Movements and Vision*, Plenum. New York., 1967.

- 
- [25] Michael I. Posner, “Orienting of attention,” *The Quarterly Journal of Experimental Psychology*, vol. 32, no. 1, pp. 3–25, 1980.
- [26] Jeremy M Wolfe, Serena J Butcher, Carol Lee, Megan Hyle, et al., “Changing your mind: On the contributions of top-down and bottom-up guidance in visual search for feature singletons,” *Journal of Experimental Psychology-Human Perception and Performance*, vol. 29, no. 2, pp. 483–501, 2003.
- [27] Keith Rayner, “Eye movements in reading and information processing: 20 years of research,” *Psychological bulletin*, vol. 124, no. 3, pp. 372, 1998.
- [28] Alfred L Yarbus, *Eye movements during perception of complex objects*, Springer, 1967.
- [29] Xiaodi Hou and Liqing Zhang, “Saliency detection: A spectral residual approach,” in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1–8.
- [30] Neil DB Bruce and John K Tsotsos, “Saliency, attention, and visual search: An information theoretic approach,” *Journal of vision*, vol. 9, no. 3, 2009.
- [31] Ronald A. Rensink, J. Kevin O’Regan, and James J. Clark, “To see or not to see: The need for attention to perceive changes in scenes,” *Psychological Science*, vol. 8, no. 5, pp. 368–373, 1997.
- [32] A. Toet, “Computational versus psychophysical bottom-up image saliency: A comparative evaluation study,” *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, vol. 33, pp. 2131–2146, Nov. 2011.
- [33] Xinyi Cui, Qingshan Liu, and Dimitris Metaxas, “Temporal spectral residual: Fast motion saliency detection,” in *Proceedings of the 17th ACM International Conference on Multimedia*, New York, NY, USA, 2009, MM ’09, pp. 617–620, ACM.
- [34] Haroon Qureshi, “DCT based temporal image signature approach,” in *VISAPP 2013 - Proceedings of the International Conference on Computer Vision Theory and Applications, Volume 1, Barcelona, Spain, 21-24 February, 2013.*, 2013, pp. 208–212.
- [35] Xiaodi Hou, J. Harel, and C. Koch, “Image signature: Highlighting sparse salient regions,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 1, pp. 194–201, Jan 2012.
- [36] A. Maki, P. Nordlund, and J.-O. Eklundh, “A computational model of depth-based attention,” in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 1996, vol. 4, pp. 734–739.

- 
- [37] N. Ouerhani and H. Hugli, “Computing visual attention from scene depth,” in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 2000, vol. 1, pp. 375–378.
- [38] Yun Zhang, Gangyi Jiang, Mei Yu, and Ken Chen, “Stereoscopic visual attention model for 3D video,” in *Proceedings of the 16th international conference on Advances in Multimedia Modeling*, Berlin, Heidelberg, 2010, MMM’10, pp. 314–324, Springer-Verlag.
- [39] Ekaterina Potapova, Michael Zillich, and Markus Vincze, “Learning what matters: combining probabilistic models of 2D and 3D saliency cues,” in *Computer Vision Systems*, pp. 132–142. Springer, 2011.
- [40] Qiuping Jiang, Fenfang Duan, and Feng Shao, “3D visual attention for stereoscopic image quality assessment,” *Journal of Software*, vol. 9, no. 7, pp. 1841–1847, 2014.
- [41] I. Iatsun, M.-C. Larabi, and C. Fernandez-Maloigne, “Spatio-temporal modeling of visual attention for stereoscopic 3D video,” in *Image Processing (ICIP) 2014 IEEE International Conference on*, Oct. 2014, pp. 5397–5401.
- [42] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau, “A coherent computational approach to model bottom-up visual attention,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 5, pp. 802–817, May 2006.
- [43] Gert Kootstra, Bart de Boer, and Lambert RB Schomaker, “Predicting eye fixations on complex visual stimuli using local symmetry,” *Cognitive computation*, vol. 3, no. 1, pp. 223–240, 2011.
- [44] Qi Zhao and Christof Koch, “Learning a saliency map using fixated locations in natural scenes,” *Journal of vision*, vol. 11, no. 3, pp. 9, 2011.
- [45] Robert J. Peters, Asha Iyer, Laurent Itti, and Christof Koch, “Components of bottom-up gaze allocation in natural images,” *Vision Research*, vol. 45, no. 18, pp. 2397 – 2416, 2005.
- [46] A. Borji and L. Itti, “State-of-the-art in visual attention modeling,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 185–207, Jan. 2013.
- [47] Ruth Rosenholtz, Amal Dorai, and Rosalind Freeman, “Do predictions of visual perception aid design?,” *ACM Trans. Appl. Percept.*, vol. 8, no. 2, pp. 12:1–12:20, Feb. 2011.
- [48] Byung-Woo Hong and M. Brady, “Segmentation of mammograms in topographic approach,” in *Visual Information Engineering, 2003. VIE 2003. International Conference on*, Jul. 2003, pp. 157–160.

- 
- [49] N Parikh, L Itti, and J Weiland, “Saliency-based image processing for retinal prostheses,” *Journal of Neural Engineering*, vol. 7, no. 1, pp. 016006, 2010.
- [50] AJAY K. MISHRA and YIANNIS ALOIMONOS, “Active segmentation,” *International Journal of Humanoid Robotics*, vol. 06, no. 03, pp. 361–386, 2009.
- [51] Qi Ma, Liming Zhang, and Bin Wang, “New strategy for image and video quality assessment,” *Journal of Electronic Imaging*, vol. 19, no. 1, pp. 011019–011019–14, 2010.
- [52] L. Itti, “Automatic foveation for video compression using a neurobiological model of visual attention,” *Image Processing, IEEE Transactions on*, vol. 13, no. 10, pp. 1304–1318, Oct. 2004.
- [53] Chenlei Guo and Liming Zhang, “A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression,” *Image Processing, IEEE Transactions on*, vol. 19, no. 1, pp. 185–198, Jan. 2010.
- [54] Sunhyoung Han and Nuno Vasconcelos, “Biologically plausible saliency mechanisms improve feedforward object recognition,” *Vision Research*, vol. 50, no. 22, pp. 2295 – 2307, 2010, Mathematical Models of Visual Coding.
- [55] Wen-Huang Cheng, Wei-Ta Chu, Jin-Hau Kuo, and Ja-Ling Wu, “Automatic video region-of-interest determination based on user attention model,” in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, 2005, vol. 4, pp. 3219–3222.
- [56] C.M. Privitera and L.W. Stark, “Algorithms for defining visual regions-of-interest: comparison with eye fixations,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 9, pp. 970–982, 2000.
- [57] Jing Zhang, Li Zhuo, and Yingdi Zhao, “Region of interest detection based on visual perception model,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, no. 02, pp. 1255005, 2012.
- [58] C. Siagian and L. Itti, “Biologically inspired mobile robot vision localization,” *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 861–873, 2009.
- [59] Yu-Fei Ma, Xian-Sheng Hua, Lie Lu, and Hong-Jiang Zhang, “A generic framework of user attention model and its application in video summarization,” *Multimedia, IEEE Transactions on*, vol. 7, no. 5, pp. 907–919, 2005.
- [60] Sophie Marat, Mickaël Guironnet, Denis Pellerin, et al., “Video summarization using a visual attention model,” in *Proceedings of the 15th European Signal Processing Conference, EUSIPCO-2007*, 2007.

- [61] Sophie Marat, Tien Ho Phuoc, Lionel Granjon, Nathalie Guyader, Denis Pellerin, and Anne Guérin-Dugué, “Modelling spatio-temporal saliency to predict gaze direction for short videos,” *International Journal of Computer Vision*, vol. 82, no. 3, pp. 231–243, 2009.
- [62] Jie-Ling Lai and Yang Yi, “Key frame extraction based on visual attention model,” *Journal of Visual Communication and Image Representation*, vol. 23, no. 1, pp. 114–125, 2012.
- [63] Naveed Ejaz, Irfan Mehmood, and Sung Wook Baik, “Efficient visual attention based framework for extracting key frames from videos,” *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 34–44, 2013.
- [64] Naveed Ejaz, Irfan Mehmood, and Sung Wook Baik, “Feature aggregation based visual attention model for video summarization,” *Computers & Electrical Engineering*, vol. 40, no. 3, pp. 993–1005, 2014.
- [65] G. Boccignone, A. Chianese, V. Moscato, and A. Picariello, “Foveated shot detection for video segmentation,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 3, pp. 365–377, 2005.
- [66] V. A. Mateescu and I. V. Bajic, “Visual attention retargeting,” *IEEE MultiMedia*, vol. 23, no. 1, pp. 82–91, Jan. 2016.
- [67] L. Wolf, M. Guttman, and D. Cohen-Or, “Non-homogeneous content-driven video-retargeting,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct. 2007, pp. 1–6.
- [68] Michael Rubinstein, Ariel Shamir, and Shai Avidan, “Improved seam carving for video retargeting,” *ACM Trans. Graph.*, vol. 27, no. 3, pp. 16:1–16:9, Aug. 2008.
- [69] Stephan Kopf, Johannes Kiess, Hendrik Lemelson, and Wolfgang Effelsberg, “Fscav: Fast seam carving for size adaptation of videos,” in *Proceedings of the 17th ACM International Conference on Multimedia*, New York, NY, USA, 2009, MM ’09, pp. 321–330, ACM.
- [70] M. Grundmann, V. Kwatra, Mei Han, and I. Essa, “Discontinuous seam-carving for video retargeting,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, Jun. 2010, pp. 569–576.
- [71] Xin Fan, Xing Xie, He-Qin Zhou, and Wei-Ying Ma, “Looking into video frames on small displays,” in *Proceedings of the Eleventh ACM International Conference on Multimedia*, New York, NY, USA, 2003, MULTIMEDIA ’03, pp. 247–250, ACM.
- [72] Jun Wang, M.J.T. Reinders, R.L. Lagendijk, J. Lindenberg, and M.S. Kankanhalli, “Video content representation on tiny devices,” in *Multimedia and Expo, 2004. ICME ’04. 2004 IEEE International Conference on*, Jun. 2004, vol. 3, pp. 1711–1714 Vol.3.

- [73] Stephan Kopf and Wolfgang Effelsberg, “Mobile cinema: canonical processes for video adaptation,” *Multimedia Systems*, vol. 14, no. 6, pp. 369–375, 2008.
- [74] Ye Luo, Junsong Yuan, Ping Xue, and Qi Tian, “Salient region detection and its application to video retargeting,” in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, Jul. 2011, pp. 1–6.
- [75] Shai Avidan and Ariel Shamir, “Seam carving for content-aware image resizing,” *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.
- [76] Bo Yan, Kairan Sun, and Liu Liu, “Matching-area-based seam carving for video retargeting,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 2, pp. 302–310, Feb. 2013.
- [77] Botao Wang, Hongkai Xiong, Zhiqian Ren, and Chang Wen Chen, “Deformable shape preserving video retargeting with salient curve matching,” *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 4, no. 1, pp. 82–94, Mar. 2014.
- [78] Bing Li, Ling-Yu Duan, Jinqiao Wang, Rongrong Ji, Chia-Wen Lin, and Wen Gao, “Spatiotemporal grid flow for video retargeting,” *Image Processing, IEEE Transactions on*, vol. 23, no. 4, pp. 1615–1628, Apr. 2014.
- [79] Feng Liu and Michael Gleicher, “Video retargeting: Automating pan and scan,” in *Proceedings of the 14th ACM International Conference on Multimedia*, New York, NY, USA, 2006, MM ’06, pp. 241–250, ACM.
- [80] Yu-Shuen Wang, Hui-Chih Lin, Olga Sorkine, and Tong-Yee Lee, “Motion-based video retargeting with optimized crop-and-warp,” *ACM Trans. Graph.*, vol. 29, no. 4, pp. 90:1–90:9, Jul. 2010.
- [81] Yu-Shuen Wang, Jen-Hung Hsiao, Olga Sorkine, and Tong-Yee Lee, “Scalable and coherent video resizing with per-frame optimization,” *ACM Trans. Graph.*, vol. 30, no. 4, pp. 88:1–88:8, July 2011.
- [82] Johannes Kiess, Daniel Gritzner, Benjamin Guthier, Stephan Kopf, and Wolfgang Effelsberg, “Gpu video retargeting with parallelized seamcrop,” in *Proceedings of the 5th ACM Multimedia Systems Conference*, New York, NY, USA, 2014, MMSys ’14, pp. 139–147, ACM.
- [83] “Methodology for the subjective assessment of the quality of television pictures,” ITU-R Rec. BT.500-12, 2009.
- [84] Christel Chamaret, Olivier Le Meur, Philippe Guillotel, and Jean-Claude Chevet, “How to measure the relevance of a retargeting approach?,” in *Trends and Topics in Computer Vision*, Kiriakos N. Kutulakos, Ed., vol. 6554 of *Lecture Notes in Computer Science*, pp. 156–168. Springer Berlin Heidelberg, 2012.

- 
- [85] Jiang Peng and Qin Xiao-Lin, “Keyframe-based video summary using visual attention clues,” *IEEE Multimedia*, vol. 17, no. 2, pp. 64–73, 2010.
- [86] Ying Li, Shih-Hung Lee, Chia-Hung Yeh, and C-CJ Kuo, “Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques,” *Signal Processing Magazine, IEEE*, vol. 23, no. 2, pp. 79–89, 2006.
- [87] Pedro Miguel Fonseca and Fernando Pereira, “Automatic video summarization based on MPEG-7 descriptions,” *Signal Processing: Image Communication*, vol. 19, no. 8, pp. 685 – 699, 2004.
- [88] Jurandy Almeida, Neucimar J Leite, and Ricardo da S Torres, “Vison: Video summarization for online applications,” *Pattern Recognition Letters*, vol. 33, no. 4, pp. 397–409, 2012.
- [89] Ba Tu Truong and Svetha Venkatesh, “Video abstraction: A systematic review and classification,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, vol. 3, no. 1, pp. 3, 2007.
- [90] Toshihiko Yamasaki and Kiyoharu Aizawa, “Motion segmentation and retrieval for 3D video based on modified shape distribution,” *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, pp. 211–211, Jan. 2007.
- [91] Lino Ferreira, Luís Cruz, and Pedro Amado Assunção, “3D video key-frame selection based on colour and depth,” in *Conf. on Telecommunications - ConfTele*, May 2013, pp. 165–168.
- [92] Yu-Jin Zhang, *Advances in Image and Video Segmentation*, IRM Press, USA, 2006.
- [93] Jinhui Yuan, Huiyi Wang, Lan Xiao, Wujie Zheng, Jianmin Li, Fuzong Lin, and Bo Zhang, “A formal study of shot boundary detection,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 2, pp. 168–186, Feb 2007.
- [94] Alan F. Smeaton, Paul Over, and Aiden R. Doherty, “Video shot boundary detection: Seven years of itrecvid activity,” *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 411 – 418, 2010, Special issue on Image and Video Retrieval Evaluation.
- [95] C. Cotsaces, N. Nikolaidis, and I. Pitas, “Video shot detection and condensed representation: a review,” *Signal Processing Magazine, IEEE*, vol. 23, no. 2, pp. 28–37, 2006.
- [96] JeHo Nam and A.H. Tewfik, “Detection of gradual transitions in video sequences using b-spline interpolation,” *Multimedia, IEEE Transactions on*, vol. 7, no. 4, pp. 667–679, Aug. 2005.
- [97] Shiguo Lian, “Automatic video temporal segmentation based on multiple features,” *Soft Computing*, vol. 15, pp. 469–482, 2011.



- [98] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso, "Temporal video segmentation to scenes using high-level audiovisual features," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 21, no. 8, pp. 1163–1177, Aug. 2011.
- [99] N.D. Doulamis, A.D. Doulamis, Y.S. Avrithis, K.S. Ntalianis, and S.D. Kollias, "Efficient summarization of stereoscopic video sequences," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, no. 4, pp. 501–517, Jun. 2000.
- [100] Boon-Lock Yeo and B. Liu, "Rapid scene analysis on compressed video," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 5, no. 6, pp. 533–544, Dec. 1995.
- [101] K. Papachristou, A. Tefas, N. Nikolaidis, and I. Pitas, "Stereoscopic video shot classification based on weighted linear discriminant analysis," in *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*, Sep. 2014, pp. 1–6.
- [102] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/fade detection and video summarization," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 1, pp. 82–91, Jan. 2006.
- [103] P.J. Besl and Neil D. McKay, "A method for registration of 3-D shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [104] Bogdan Ionescu, Didier Coquin, Patrick Lambert, and Vasile Buzuloiu, "A fuzzy color-based approach for understanding animated movies content in the indexing task," *Eurasip Journal on Image and Video Processing*, vol. 10, no. 2008, pp. 1–17, 2008.
- [105] W.A.C. Fernando, C.N. Canagarajah, and D.R. Bull, "Fade and dissolve detection in uncompressed and compressed video sequences," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, 1999, vol. 3, pp. 299–303.
- [106] Rim Slama, Hazem Wannous, and Mohamed Daoudi, "3D human motion analysis framework for shape similarity and retrieval," *Image and Vision Computing*, vol. 32, no. 2, pp. 131–154, 2014.
- [107] U. Gargi, R. Kasturi, and S.H. Strayer, "Performance characterization of video-shot-change detection methods," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, no. 1, pp. 1–13, Feb. 2000.
- [108] Yiming Yang and Xin Liu, "A re-examination of text categorization methods," in *Proceedings of the 22<sup>nd</sup> annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, Aug. 1999, SIGIR'99, pp. 42–49, ACM.

- 
- [109] Rui Xu and II Wunsch, D., “Survey of clustering algorithms,” *Neural Networks, IEEE Transactions on*, vol. 16, no. 3, pp. 645–678, May 2005.
- [110] Ke-Sen Huang, Chun-Fa Chang, Yu-Yao Hsu, and Shi-Nine Yang, “Key probe: a technique for animation keyframe extraction,” *The Visual Computer*, vol. 21, no. 8-10, pp. 532–541, 2005.
- [111] Chao Jin, Thomas Fevens, and Sudhir Mudur, “Optimized keyframe extraction for 3D character animations,” *Computer Animation and Virtual Worlds*, vol. 23, no. 6, pp. 559–568, 2012.
- [112] Lino Ferreira, Luis A. da Silva Cruz, and Pedro Assuncao, “A generic framework for optimal 2D/3D key-frame extraction driven by aggregated saliency maps,” *Signal Processing: Image Communication*, vol. 39, Part A, pp. 98 – 110, 2015.
- [113] Yanwei Fu, Yanwen Guo, Yanshu Zhu, Feng Liu, Chuanming Song, and Zhi-Hua Zhou, “Multi-view video summarization,” *Multimedia, IEEE Transactions on*, vol. 12, no. 7, pp. 717–729, 2010.
- [114] Sandra Eliza Fontes Avila, Ana Paula Brandão Lopes, et al., “Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method,” *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.
- [115] Daniel DeMenthon, Vikrant Kobla, and David Doermann, “Video summarization by curve simplification,” in *Proceedings of the Sixth ACM International Conference on Multimedia*, New York, NY, USA, 1998, MULTIMEDIA ’98, pp. 211–218, ACM.
- [116] L.J. Latecki, D. de Wildt, and Jianying Hu, “Extraction of key frames from videos by optimal color composition matching and polygon simplification,” in *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*, 2001, pp. 245–250.
- [117] J. Calic and E. Izquierdo, “Efficient key-frame extraction and video analysis,” in *Information Technology: Coding and Computing, 2002. Proceedings. International Conference on*, Apr. 2002, pp. 28–33.
- [118] Ik Soo Lim and D. Thalmann, “Key-posture extraction out of human motion data,” in *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, 2001, vol. 2, pp. 1167–1169.
- [119] N.D. Doulamis, A.D. Doulamis, Y. Avrithis, and S.D. Kollias, “A stochastic framework for optimal key frame extraction from MPEG video databases,” in *Multimedia Signal Processing, 1999 IEEE 3rd Workshop on*, 1999, pp. 141–146.
- [120] Byeong-Doo Choi, Jong-Woo Han, Chang-Su Kim, and Sung-Jea Ko, “Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 4, pp. 407–416, Apr. 2007.

- 
- [121] Jianfeng Xu, Toshihiko Yamasaki, and Kiyoharu Aizawa, “Summarization of 3D video by rate-distortion trade-off,” *IEICE Transactions on Information and Systems*, vol. E90-D, pp. 1430–1438, 2007.
- [122] Peng Huang, Adrian Hilton, and Jonathan Starck, “Automatic 3D Video Summarization: Key Frame Extraction from Self-Similarity,” in *3DPVT '08: Proceedings of the Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, Washington, DC, USA, 2008, IEEE Computer Society.
- [123] Yihong Gong and Xin Liu, “Video summarization using singular value decomposition,” in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 2000, vol. 2, pp. 174–180.
- [124] M. Cooper and J. Foote, “Summarizing video using non-negative similarity matrix factorization,” in *Multimedia Signal Processing, 2002 IEEE Workshop on*, Dec. 2002, pp. 25–28.
- [125] W. Abd-Almageed, “Online, simultaneous shot boundary detection and key frame extraction for sports videos using rank tracing,” in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct. 2008, pp. 3200–3203.
- [126] Tong-Yee Lee, Chao-Hung Lin, Yu-Shuen Wang, and Tai-Guang Chen, “Animation key-frame extraction and simplification using deformation analysis,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 4, pp. 478–486, Apr. 2008.
- [127] Jackie Assa, Yaron Caspi, and Daniel Cohen-Or, “Action synopsis: pose selection and illustration,” *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 667–676, 2005.
- [128] Chang Ha Lee, Amitabh Varshney, and David W. Jacobs, “Mesh saliency,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 659–666, Jul. 2005.
- [129] Cihan Halit and Tolga Capin, “Multiscale motion saliency for keyframe extraction from motion capture sequences,” *Computer Animation and Virtual Worlds*, vol. 22, no. 1, pp. 3–14, 2011.
- [130] Qing Xu, Pengcheng Wang, Bin Long, M. Sbert, M. Feixas, and R. Scopigno, “Selection and 3D visualization of video key frames,” in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, Oct. 2010, pp. 52–59.
- [131] Cuong Nguyen, Yuzhen Niu, and Feng Liu, “Video summagator: An interface for video summarization and navigation,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2012, CHI '12, pp. 647–650, ACM.
- [132] ITU-R, “Methodology for the subjective assessment of the quality of television pictures - Tech. Rep. BT.500-13,” 2012.

- 
- [133] Padmavathi Mundur, Yong Rao, and Yelena Yesha, “Keyframe-based video summarization using delaunay clustering,” *International Journal on Digital Libraries*, vol. 6, no. 2, pp. 219–232, 2006.
- [134] Marco Furini, Filippo Geraci, Manuela Montangero, and Marco Pellegrini, “Stimo: Still and moving video storyboard for the web scenario,” *Multimedia Tools and Applications*, vol. 46, no. 1, pp. 47–69, 2010.
- [135] Tie-Yan Liu, Xu-Dong Zhang, Jian Feng, and Kwok-Tung Lo, “Shot reconstruction degree: a novel criterion for key frame selection,” *Pattern Recognition Letters*, pp. 1451–1457, 2004.
- [136] Hyun Sung Chang, Sanghoon Sull, and Sang Uk Lee, “Efficient video indexing scheme for content-based retrieval,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 9, no. 8, pp. 1269–1279, Dec. 1999.
- [137] Ciocca Gianluigi and Schettini Raimondo, “An innovative algorithm for key frame extraction in video summarization,” *Journal of Real-Time Image Processing*, vol. 1, pp. 69–88, 2006.
- [138] Qing-Ge Ji, Zhi-Dang Fang, Zhen-Hua Xie, and Zhe-Ming Lu, “Video abstraction based on the visual attention model and online clustering,” *Signal Processing: Image Communication*, vol. 28, no. 3, pp. 241 – 253, 2013.
- [139] Andrei Khodakovsky, Peter Schröder, and Wim Sweldens, “Progressive geometry compression,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 2000, SIGGRAPH ’00, pp. 271–278, ACM Press/Addison-Wesley Publishing Co.
- [140] Hector M. Briceño, Pedro V. Sander, Leonard McMillan, Steven Gortler, and Hugues Hoppe, “Geometry videos: A new representation for 3D animations,” in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Aire-la-Ville, Switzerland, Switzerland, 2003, SCA ’03, pp. 136–146, Eurographics Association.
- [141] Klaus Schoeffmann, Frank Hopfgartner, Oge Marques, Laszlo Boeszoermyeni, and Joemon M. Jose, “Video browsing interfaces and applications: a review,” *Journal of Photonics for Energy*, pp. 018004–018004–35, 2010.
- [142] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić, “Feature-based similarity search in 3D object databases,” *ACM Comput. Surv.*, vol. 37, no. 4, pp. 345–387, Dec. 2005.
- [143] Thibault Napoléon and Hichem Sahbi, “From 2D silhouettes to 3D object retrieval: contributions and benchmarking,” *Journal on Image and Video Processing*, vol. 2010, pp. 1, 2010.

- [144] Michalis A. Savelonas, Ioannis Pratikakis, and Konstantinos Sfikas, “An overview of partial 3D object retrieval methodologies,” *Multimedia Tools and Applications*, pp. 1–26, 2014.
- [145] Nicholas Vretos, Nikos Nikolaidis, and Ioannis Pitas, “The use of audio-visual description profile in 3D video content description,” in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, Oct. 2012, pp. 1–4.
- [146] M. Sano, W. Bailer, A. Messina, J.-P. Evain, and M. Matton, “The MPEG-7 audiovisual description profile (AVDP) and its application to multi-view video,” in *IVMSP Workshop, 2013 IEEE 11th*, Jun. 2013, pp. 1–4.
- [147] Volker Blanz and Thomas Vetter, “A morphable model for the synthesis of 3D faces,” in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1999, SIGGRAPH '99, pp. 187–194, ACM Press/Addison-Wesley Publishing Co.
- [148] Christian R. Shelton, “Morphable surface models,” *International Journal of Computer Vision*, vol. 38, no. 1, pp. 75–91, 2000.
- [149] Jianfeng Xu, T. Yamasaki, and K. Aizawa, “Motion editing in 3D video database,” in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, Jun. 2006, pp. 472–479.
- [150] Rick Parent, *Computer animation: algorithms and techniques*, Morgan-Kaufmann, USA, 2012.
- [151] Xun Cao, Zheng Li, and Qionghai Dai, “Semi-automatic 2D-to-3D conversion using disparity propagation,” *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 491–499, Jun. 2011.
- [152] W.-N. Lie, C.-Y. Chen, and W.-C. Chen, “2D to 3D video conversion with key-frame depth propagation and trilateral filtering,” *Electronics Letters*, vol. 47, no. 5, pp. 319–321, Mar. 2011.
- [153] Dichangsheng Wang, Ju Liu, Jiande Sun, Wei Liu, and Yujun Li, “A novel key-frame extraction method for semi-automatic 2D-to-3D video conversion,” in *Broadband Multimedia Systems and Broadcasting (BMSB), 2012 IEEE International Symposium on*, Jun. 2012, pp. 1–5.
- [154] Kuanyu Ju and Hongkai Xiong, “A semi-automatic 2D-to-3D video conversion with adaptive key-frame selection,” in *SPIE/COS Photonics Asia*. International Society for Optics and Photonics, 2014, pp. 92730M–92730M.
- [155] J. De Cock, S. Notebaert, P. Lambert, and R. Van De Walle, “Architectures for fast transcoding of H.264/AVC to quality-scalable svc streams,” *Multimedia, IEEE Transactions on*, vol. 11, no. 7, pp. 1209–1224, 2009.

- 
- [156] H. Schwarz, D. Marpe, T. Schierl, and T. Wiegand, “Combined scalability support for the scalable extension of H.264/AVC,” in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, 2005, pp. 4 pp.–.
- [157] G.J. Sullivan, J. Ohm, Woo-Jin Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [158] J. Ohm, G.J. Sullivan, H. Schwarz, Thiow Keng Tan, and T. Wiegand, “Comparison of the coding efficiency of video coding standards-including high efficiency video coding (HEVC),” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1669–1684, 2012.
- [159] Hyomin Choi, J. Nam, D. Sim, and I.V. Bajic, “Scalable video coding based on high efficiency video coding (HEVC),” in *Communications, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference on*, 2011, pp. 346–351.
- [160] Zhongbo Shi, Xiaoyan Sun, and Feng Wu, “Spatially scalable video coding for HEVC,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1813–1826, 2012.
- [161] Tobias Hinz, Philipp Helle, Haricharan Lakshman, Mischa Siekmann, Jan Stegmann, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, “An HEVC extension for spatial and quality scalable video coding,” in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 866605–866605.
- [162] Mei-Juan Chen, Ming-Chieh Chi, Ching-Ting Hsu, and Jeng-Wei Chen, “ROI video coding based on H.263+ with robust skin-color detection technique,” in *Consumer Electronics, 2003. ICCE. 2003 IEEE International Conference on*, 2003, pp. 44–45.
- [163] L.S. Karlsson and M. Sjoström, “Region-of-interest 3D video coding based on depth images,” in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, 2008, pp. 141–144.
- [164] T. Stockhammer, M.M. Hannuksela, and Stephan Wenger, “H.26L/JVT coding network abstraction layer and ip-based transport,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, 2002, vol. 2, pp. II–485–II–488.
- [165] Sebastiaan Van Leuven, Kris Van Schevensteen, Tim Dams, and Peter Schelkens, “An implementation of multiple region-of-interest models in H.264/AVC,” in *Signal Processing for Image Enhancement and Multimedia Processing*, pp. 215–225. Springer, 2008.
- [166] F. Peng, X. Zhu, and M. Long, “A ROI privacy protection scheme for H.264 video based on fmo and chaos,” *Information Forensics and Security, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

- 
- [167] D. Grois, E. Kaminsky, and O. Hadar, “Dynamically adjustable and scalable ROI video coding,” in *Broadband Multimedia Systems and Broadcasting (BMSB), 2010 IEEE International Symposium on*, 2010, pp. 1–5.
- [168] Jung-Hwan Lee and C. Yoo, “Scalable ROI algorithm for H.264/SVC-based video streaming,” *Consumer Electronics, IEEE Transactions on*, vol. 57, no. 2, pp. 882–887, 2011.
- [169] L. Pinto and P. Assuncao, “Asymmetric 3D video coding using regions of perceptual relevance,” in *2012 International Conference on 3D Imaging (IC3D)*, Dec. 2012, pp. 1–6.
- [170] Iain E Richardson, *H.264 and MPEG-4 video compression: video coding for next-generation multimedia*, John Wiley & Sons, 2004.
- [171] Gisle Bjontegaard, “Calculation of average PSNR differences between RD-curves,” *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April 2001*, 2001.
- [172] Lino Ferreira, Luís Cruz, and Pedro Amado Assunção, “Efficient scalable coding of video summaries using dynamic gop structures,” in *EUROCON 2011 IEEE*, Apr. 2011, pp. 1–4.
- [173] Luis Herranz and Shuqiang Jiang, “Scalable storyboards in handheld devices: applications and evaluation metrics,” *Multimedia Tools and Applications*, pp. 1–29, 2015.
- [174] Paul Over, Alan F. Smeaton, and George Awad, “The Trecvid 2008 BBC rushes summarization evaluation,” in *Proceedings of the 2Nd ACM TRECVID Video Summarization Workshop*, New York, NY, USA, 2008, TVS '08, pp. 1–20, ACM.
- [175] Paul Over, Alan F. Smeaton, and Philip Kelly, “The Trecvid 2007 BBC rushes summarization evaluation pilot,” in *Proceedings of the International Workshop on TRECVID Video Summarization*, New York, NY, USA, 2007, TVS '07, pp. 1–15, ACM.
- [176] J. Bescos, J.M. Martinez, L. Herranz, and F. Tiburzi, “Content-driven adaptation of on-line video,” in *Content-Based Multimedia Indexing, 2007. CBMI '07. International Workshop on*, 2007, pp. 122–129.
- [177] M. Mrak, J. Calic, and A. Kondoz, “Fast analysis of scalable video for adaptive browsing interfaces,” *Comp. Vision and Image Understanding*, vol. 113, no. 3, pp. 425–434, 2009.
- [178] L. Herranz and J. Martinez, “An integrated approach to summarization and adaptation using H.264/MPEG-4 SVC,” *Signal Processing: Image Comm.*, vol. 24, no. 6, pp. 499 – 509, 2009.

- 
- [179] Hugo Boujut, Jenny Benois-Pineau, and Remi Megret, “Fusion of multiple visual cues for visual saliency extraction from wearable camera settings with strong motion,” in *Computer Vision-ECCV. Workshops and Demonstrations*. Springer, 2012, pp. 436–445.
- [180] Allen Brookes and Kent A Stevens, “The analogy between stereo depth and brightness,” *Perception*, vol. 18, pp. 601–614, 1989.
- [181] Dwarikanath Mahapatra, Stefan Winkler, and Shih-Cheng Yen, “Motion saliency outweighs other low-level features while watching videos,” in *Proc. SPIE Human Vision and Electronic Imaging XIII*, 2008, vol. 6806, p. 68060P.
- [182] H. Kim, Sanghoon Lee, and A.C. Bovik, “Saliency prediction on stereoscopic videos,” *IEEE Trans. on Image Processing*, vol. 23, pp. 1476–1490, Apr. 2014.
- [183] Lijuan Duan, Chunpeng Wu, Jun Miao, and A.C. Bovik, “Visual conspicuity index: Spatial dissimilarity, distance, and central bias,” *Signal Processing Letters, IEEE*, vol. 18, no. 11, pp. 690–693, Nov. 2011.
- [184] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1, pp. I-511–I-518.
- [185] P.M. Mather and M. Koch, *Computer Processing of Remotely-Sensed Images: An Introduction*, Wiley, 2011.
- [186] Joint Video Team and ITU-T Video Coding Experts Group, “JSVM software manual,” Jun. 2007.
- [187] Stefano Mattoccia, Simone Giardino, and Andrea Gambini, “Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering,” *Computer Vision-ACCV 2009*, pp. 371–380, 2010.
- [188] Dimitri P. Bertsekas, *Dynamic programming: deterministic and stochastic models*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1987.
- [189] Jiachen Yang, Chunping Hou, Yuan Zhou, Zhuoyun Zhang, and Jichang Guo, “Objective quality assessment method of stereo images,” in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, May 2009, pp. 1–4.
- [190] Chaminda TER Hewage, Maria G Martini, and Harsha D Appuhami, “ROI-based transmission method for stereoscopic video to maximize rendered 3D video quality,” in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2012.



- 
- [191] P. Lebreton, A. Raake, M. Barkowsky, and P. Le Callet, “Evaluating depth perception of 3D stereoscopic videos,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 6, no. 6, pp. 710–720, Oct. 2012.
- [192] John Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [193] S.L.P. Yasakethu, D.V.S.X. De Silva, W.A.C. Fernando, and A. Kondoz, “Predicting sensation of depth in 3D video,” *Electronics Letters*, vol. 46, no. 12, pp. 837–839, 2010.
- [194] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.
- [195] Olivier Le Meur and Zhi Liu, “Saliency aggregation: Does unity make strength?,” in *Computer Vision - ACCV 2014*, Daniel Cremers, Ian Reid, Hideo Saito, and Ming-Hsuan Yang, Eds., vol. 9006 of *Lecture Notes in Computer Science*, pp. 18–32. Springer International Publishing, 2015.
- [196] G. T. Buswell, *How people look at pictures: a study of the psychology and perception in art*, Univ. Chicago Press., Chicago, USA, 1 edition, 1935.
- [197] Vincent Buso, Jenny Benois-Pineau, and Jean-Philippe Domenger, “Geometrical cues in visual saliency models for active object recognition in egocentric videos,” in *Proc. of the 1st International Workshop on Perception Inspired Video Processing*. 2014, PIVP ’14, pp. 9–14, ACM.
- [198] Gary Marchionini, Barbara M. Wildemuth, Gary Geisler, and Yaxiao Song, “The Open Video Project,” <http://www.open-video.org/>, Last accessed: 2016-04-10.
- [199] M. Urvoy, M. Barkowsky, R. Cousseau, Y. Koudota, V. Ricorde, P. Le Callet, J. Gutierrez, and N. Garcia, “NAMA3DS1-COSPAD1: Subjective video quality assessment database on coding conditions introducing freely available high quality 3D stereoscopic sequences,” in *Quality of Multimedia Experience (QoMEX), Fourth International Workshop on*, Jul. 2012, pp. 109–114.
- [200] ISO/IEC JTC1/SC29/WG11, “Proposed stereo test sequences for 3D video coding,” 2012.
- [201] ISO/IEC JTC1/SC29/WG11, “Poznan multiview video test sequences and camera parameter,” 2009.
- [202] Dwarikanath Mahapatra, Stefan Winkler, and Shih-Cheng Yen, “Motion saliency outweighs other low-level features while watching videos,” in *Proc. SPIE Human Vision and Electronic Imaging XIII*, 2008, vol. 6806, p. 68060P.

- [203] David S. Wooding, “Fixation maps: Quantifying eye-movement traces,” in *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, New York, NY, USA, 2002, ETRA '02, pp. 31–36, ACM.
- [204] J. Lee and H. Kalva, *The VC-1 and H.264 Video Compression Standards for Broadband Video Services*, Berlin, springer us edition, 2008.
- [205] C. Chamaret, J. C. Chevet, and O. Le Meur, “Spatio-temporal combination of saliency maps and eye-tracking assessment of different strategies,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sept 2010, pp. 1077–1080.