N° d'ordre : 2013ISAL0086 Année 2013

Institut National des Sciences Appliquées de Lyon Université de Passau

École doctorale InfoMaths : Informatique et Mathématiques (EDA 512)

Semantic Protection and Personalization of Video Content. PIAF: MPEG compliant adaptation framework preserving the user perceived quality

Thesis

Presented and publicly defended in the 23th of September 2013

To obtain

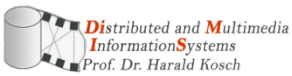
The degree of Doctor in Computer Science

by

Vanessa El-Khoury

Committee members

Rapporteurs:	Aris OUKSEL	Professor (University of Illinois at Chicago)
	Vincent CHARVILLAT	Professor (Institut National Polytechnique de Toulouse)
Examiners:	Dietmar JANNACH	Professor (Technische Universität Dortmund)
	Gordon BLAIR	Professor (School of Computing and Communications)
	Michael GRANITZER	Professor (Universität Passau)
	David COQUIL	Invited Researcher (Universität Passau)
Supervisors:	Lionel BRUNIE	Professor (LIRIS-INSA Lyon)
	Harald KOSCH	Professor (Universität Passau)
	Nadia BENNANI	Maître de Conférences (LIRIS-INSA Lyon)



Laboratoire d'InfoRmatique en Images et Systèmes d'information - UMR 5205

Lehrstuhl für Verteilte Informationssysteme

"Do not go where the path may lead, go instead where there is no path and leave a trail." - Ralph Waldo Emerson -

thus is research ...

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisors Prof. Lionel Brunie, Prof. Harald Kosch, Dr. Nadia Bennani and Dr. David Coquil for their continuous support of my PhD study and research, for their patience and tenacity over the last five years. Their guidance and immense knowledge helped me in all the time of research and writing of this thesis. Thank you for supporting me till the finishing line of my PhD study.

I would also like to say thanks the Université Franco-Allemande (UFA) for the financial support and for hosting the defense of my thesis in Saarbrücken. I would especially like to thank Maria Leprévost and Kathleen Schlütter.

I would also like to thank the members of my doctoral committee : Prof. Aris Ouksel, Prof. Vincent Charvillat, Prof. Dietmar Jannach, Prof. Gordon Blair and Prof. Michael Granitzer, for their input and insightful comments.

My sincere thanks goes again to Dr. David Coquil who generously gave up his time for me not only to make scientific discussion but also to help me to integrate into the German society easily. I would like to extend my sincerest thanks to Dr. Gabriele Gianini for his valuable feedbacks.

My thanks also go to both secretaries in Lyon and Passau : Mrs. Mabrouka Gheraissa and Mrs. Ingrid Winter, for helping me in the administration tasks. You are the best secretaries ever !

I also thank my fellow labmates of both research groups in the "Lehrstul für verteilte Informationssysteme" of Pr. Kosch at the university of Passau (Germany), and in the DRIM team of Pr. Brunie in LIRIS laboratory at INSA de Lyon (France) for their support : Armelle Natacha Ndjafa, Andreas Wölfl, Britta Meixner, Stella Maria Stars, Christian Vilsmaier, Florian Stegmaier, Getnet Abebe, Hatem Mousselly Sergieh, Lyes Limam, Merza Klaghstan, Pr. Mario Döller, Dr. Omar Hasan, Dr. Pierre-Edouard Portier, Dr. Tilmann Rabl, Dr. Tobias Mayer, Yulian Yang and Dr. Zeina Torbey.

Also I thank my friends in France, Germany and Lebanon : Anna Dokter, Bilge Sipal, Cynthia and Chadi Madi, Cedric Bertrand-Royer, Doris Khalil, Gabi and Franti Okupny, Iman Sassine, Martina and Klaus Hoffmann, Dr. Larissa Yaroslavtseva, Maria Weidinger, Michel El-Zoghby, Mireille Zaibak, Olga Ivanova, Rana Saad and Salim Abi Saber. In particular, a special thanks to Dr. Pascal Bou Nassar for all his support and motivation along the way to finish my thesis.

I would also like to thank my parents Norma and Maroun, my two brothers Nizar and Elie, and all my family in Lebanon. They were always supporting me and encouraging me with their best wishes and prayers.

Finally, a special thanks from the heart to my angel Martin Hoffmann for his moral support to keep me going till the end of the thesis. He was always there cheering me up and stood by me through the good and bad times.

Résumé

Le concept de "Universal Multimedia Experience" (Universal Multimedia Experience (UME), Expérience Multimédia Universelle) a pour but de garantir à chaque utilisateur d'un système multimédia l'accès à tout moment et en tout lieu à un contenu informatif personnalisé grâce à l'adaptation du contenu. Pour ce faire, les préférences de l'utilisateur doivent être transcrites en un ensemble de contraintes techniques ou sémantiques appliquées lors du processus d'adaptation. La prise en compte des contraintes sémantiques nécessite d'effectuer des actions sur le contenu de la vidéo à des niveaux de granularité différents allant du shot jusqu'aux objets présents sur une séquence vidéo. Dans la littérature, seuls les problèmes liés aux contraintes techniques ont été abondamment traités. Par ailleurs, d'autres contraintes liées à la propriété intellectuelle peuvent contraindre l'adaptation en limitant le champ d'action possible. Ces contraintes ont jusqu'à maintenant été négligées.

Nous proposons donc dans cette thèse un framework d'adaptation appelé "Personalized vIdeo Adaptation Framework" (Personalized vIdeo Adaptation Framework (PIAF)) conçu à partir des standards MPEG. PIAF intègre les contraintes sémantiques et vise à maximiser la qualité perçue par l'utilisateur lors de la visualisation de la vidéo tout en respectant les droits de propriété intellectuelle.

La qualité requise est fonction du degré de satisfaction de l'utilisateur dans la perception du contenu (qualité perceptuelle du contenu adapté), de la quantité d'information qui lui est fournie (qualité sémantique du contenu adapté), et du temps d'exécution du processus d'adaptation (efficacité de l'adaptation). Dans le framework d'adaptation proposé, le processus d'adaptation est contrôlé par un Moteur de Prise de Décision et d'Adaptation (MPDA). La tâche du MPDA est de produire différents plans d'adaptation en fonction des contraintes sémantiques, techniques et de qualité, puis de sélectionner le plan à mettre en oeuvre afin de maximiser la qualité. Pour cela, le MPDA doit être en mesure de relever trois challenges : (1) mesurer quantitativement la qualité de la vidéo produite par un plan d'adaptation (2) choisir parmi les plans d'adaptation celui qui produira la meilleure qualité (3) résoudre les conflits entre contraintes, notamment dans le cas où les préférences de l'utilisateur entrent en conflit avec les conditions d'adaptation de la vidéo fixées par le propriétaire (le détenteur des droits de propriété intellectuelle).

Les contributions de cette thèse peuvent être résumées comme suit. Dans un premier temps, nous avons utilisé et étendu les standards MPEG-7 et MPEG-21 afin de représenter les préférences des utilisateurs. Nous avons ensuite proposé un modèle formel du processus d'adaptation sémantique d'une vidéo et défini une fonction d'utilité régissant le mécanisme de prise de décision du MPDA. Cette fonction tient compte de différentes dimensions de qualité (qualité perceptuelle, sémantique, temps d'exécution nécessaire) afin d'évaluer quantitativement la qualité d'un plan d'adaptation. Le processus d'adaptation que nous proposons intègre les droits de propriété intellectuelle dans le processus de décision. Dans certains cas, le plan d'adaptation qui produirait la vidéo de meilleure qualité adaptée aux préférences de l'utilisateur peut être inap-

plicable car il ne respecte pas les contraintes du propriétaire. Trouver le meilleur plan d'adaptation devient alors un problème NP-complet ; nous montrons que ce problème peut se ramener à un problème d'optimisation connu.

Afin d'implémenter ce framework, nous avons également développé un outil d'annotation sémantique de contenu vidéo (Semantic Video Content Annotation Tool (SVCAT)) qui produit des annotations sémantiques structurelles et de haut niveau selon un modèle objet basé sur du contenu vidéo. Nous avons validé nos travaux avec des évaluations qualitatives et quantitatives qui nous ont permis d'étudier la performance et l'efficacité du MPDA. Les résultats obtenus démontrent que la fonction d'utilité proposée présente une forte corrélation avec les évaluations subjectives fournies par des utilisateurs concernant la qualité d'une vidéo adaptée, et constitue donc une base tout à fait pertinente pour le MPDA.

Mots-clés: Expérience Multimédia, Personnalisation des Vidéos, Adaptation Sémantique, MPEG-7, MPEG-21, Adaptation du Contenu, Modèle de Fonction d'Utilité, Qualité d'Expérience, Perception des Utilisateurs

Kurzfassung

Der Begriff "Universal Multimedia Experience" (UME) beschreibt die Vision, dass ein Nutzer nach seinen individuellen Vorlieben zugeschnittene Videoinhalte konsumieren kann. In dieser Dissertation werden im UME nun auch semantische Constraints berücksichtigt, welche direkt mit der Konsumierung der Videoinhalte verbunden sind. Dabei soll die Qualität der Videoerfahrung für den Nutzer maximiert werden. Diese Qualität ist in der Dissertation durch die Benutzerzufriedenheit bei der Wahrnehmung der Veränderung der Videos repräsentiert. Die Veränderung der Videos wird durch eine Videoadaptierung erzeugt, z.B. durch die Löschung oder Veränderung von Szenen, Objekten, welche einem semantischen Constraints nicht entsprechen.

Kern der Videoadaptierung ist die "Adaptation Decision Taking Engine" (ADTE). Sie bestimmt die Operatoren, welche die semantischen Constraints auflösen, und berechnet dann mögliche Adaptierungspläne, die auf dem Video angewandt werden sollen. Weiterhin muss die ADTE für jeden Adaptierungsschritt anhand der Operatoren bestimmen, wie die Vorlieben des Nutzers berücksichtigt werden können. Die zweite Herausforderung ist die Beurteilung und Maximierung der Qualität eines adaptierten Videos. Die dritte Herausforderung ist die Berücksichtigung sich widersprechender semantischer Constraints. Dies betrifft insbesondere solche, die mit Urheberrechten in Verbindung stehen.

In dieser Dissertation werden die oben genannten Herausforderungen mit Hilfe eines "Personalized video Adaptation Framework" (PIAF) gelöst, welche auf den "Moving Picture Expert Group" (MPEG)-Standard MPEG-7 und MPEG-21 basieren. PIAF ist ein Framework, welches den gesamten Prozess der Videoadaptierung umfasst. Es modelliert den Zusammenhang zwischen den Adaptierungsoperatoren, den Vorlieben der Nutzer und der Qualität der Videos. Weiterhin wird das Problem der optimalen Auswahl eines Adaptierungsplans für die maximale Qualität der Videos untersucht. Dafür wird eine Utility Funktion (UF) definiert und in der ADTE eingesetzt, welche die semantischen Constraints mit den vom Nutzer ausgedrückten Vorlieben vereint.

Weiterhin ist das "Semantic Video Content Annotation Tool" (SVCAT) entwickelt worden, um strukturelle und semantische Annotationen durchzuführen. Ebenso sind die Vorlieben der Nutzer mit MPEG-7 und MPEG-21 Deskriptoren berücksichtigt worden. Die Entwicklung dieser Software-Werkzeuge und Algorithmen ist notwendig, um ein vollständiges und modulares Framework zu erhalten. Dadurch deckt PIAF den kompletten Bereich der semantischen Videoadaptierung ab.

Das ADTE ist in qualitativen und quantitativen Evaluationen validiert worden. Die Ergebnisse der Evaluation zeigen unter anderem, dass die UF im Bereich Qualität eine hohe Korrelation mit der subjektiven Wahrnehmung von ausgewählten Nutzern aufweist.

Schluesselworte: Multimedia Experience, Personalisierung von Videos, semantische

Adaptierung, MPEG-7, MPEG-21, semantische Constraints, Inhaltsadaptierung, Nutzenfunktionen, Quality of Experience, Nutzerwahrnehmung.

Abstract

UME is the notion that a user should receive informative adapted content anytime and anywhere. Personalization of videos, which adapts their content according to user preferences, is a vital aspect of achieving the UME vision. User preferences can be translated into several types of constraints that must be considered by the adaptation process, including semantic constraints directly related to the content of the video. To deal with these semantic constraints, a fine-grained adaptation, which can go down to the level of video objects, is necessary. The overall goal of this adaptation process is to provide users with adapted content that maximizes their Quality of Experience (QoE). This QoE depends at the same time on the level of the user's satisfaction in perceiving the adapted content, the amount of knowledge assimilated by the user, and the adaptation execution time.

In video adaptation frameworks, the Adaptation Decision Taking Engine (ADTE), which can be considered as the "brain" of the adaptation engine, is responsible for achieving this goal. The task of the ADTE is challenging as many adaptation operations can satisfy the same semantic constraint, and thus arising in several feasible adaptation plans. Indeed, for each entity undergoing the adaptation process, the ADTE must decide on the adequate adaptation operator that satisfies the user's preferences while maximizing his/her quality of experience. The first challenge to achieve in this is to objectively measure the quality of the adapted video, taking into consideration the multiple aspects of the QoE. The second challenge is to assess beforehand this quality in order to choose the most appropriate adaptation plan among all possible plans. The third challenge is to resolve conflicting or overlapping semantic constraints, in particular conflicts arising from constraints expressed by owner's intellectual property rights about the modification of the content.

In this thesis, we tackled the aforementioned challenges by proposing a Utility Function (UF), which integrates semantic concerns with user's perceptual considerations. This UF models the relationships among adaptation operations, user preferences, and the quality of the video content. We integrated this UF into an ADTE. This ADTE performs a multi-level piecewise reasoning to choose the adaptation plan that maximizes the user-perceived quality. Furthermore, we included intellectual property rights in the adaptation process. Thereby, we modeled content owner constraints. We dealt with the problem of conflicting user and owner constraints by mapping it to a known optimization problem. Moreover, we developed the SVCAT, which produces structural and high-level semantic annotation according to an original object-based video content model. We modeled as well the user's preferences proposing extensions to Moving Picture Expert Group (MPEG)-7 and MPEG-21. All the developed contributions were carried out as part of a coherent framework called PIAF. PIAF is a complete modular MPEG standard compliant framework that covers the whole process of semantic video adaptation.

We validated this research with qualitative and quantitative evaluations, which assess the performance and the efficiency of the proposed adaptation decision-taking

engine within PIAF. The experimental results show that the proposed UF has a high correlation with subjective video quality evaluation.

Keywords: Universal Multimedia Experience, Personalization of Video, Semantic Adaptation, MPEG-7, MPEG-21, Semantic Constraint, Content Adaptation, Utility Functon Model, Quality of Experience, User Perception

Contents

Acronyms

ADTE	Adaptation Decision Taking Engine
AEE	Adaptation Execution Engine
AV	audio-visual
AVI	Audio Video Interleaved
AI	Artificial Intelligence
BSD	Bitstream Syntax Description
CATs	Content Adaptation Tools
CC/PP	Composite Capability/Preference Profiles
codec	CODer- DECoder
CI	Confidence Interval
CS	Classification Schemes
CV	Cross-Validation
D	Descriptors
DCD	Dominant Color Descriptor
DS	Description Schemes
DI	Digital Item
DIA	Digital Item Adaptation
DIP	Digital Item Processing
DDL	Description Definition Language
fps	Frame Per Second
GUB	Generalized Upper Bound
GUI	Graphical User Interface
HVS	Human Visual System
ISO/	International Organization for Standardization
IEC	International Electrotechnical Commission
JAXB	Java Architecture for XML Binding
KP	Knapsack Problem
LOO	Leave-One-Out
LOLOCV	Leave-One-label-Out Cross-Validation
LOOCV	Leave-One-Out Cross-Validation
MCKP	Multiple-Choice Knapsack Problem

MDS	Multimedia Description Schemes
MOS	Mean Opinion Score
MPEG	Moving Picture Expert Group
MSE	Mean Squared Error
ofs	object frame sequence
PIAF	Personalized vIdeo Adaptation Framework
pixel	Picture Element
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
QoP	Quality of Perception
QoS	Quality of Service
RDF	Resource Description Framework
ROI	Region Of Interest
RMSE	Root Mean Square Error
SAF	Semantic Adaptation Framework
SNR	Signal-to-Noise Ratio
SPARQL	Simple Protocol and RDF Query Language
gBSDs	generic Bitstream Syntax Descriptions
AQoSs	Adaptation Quality of Services
SQ	Satisfaction of the Overall Adapted Quality
SemPQ	Semantic Perceived Quality
SOFS	Shot Object Frame Sequence
SPQ	Spatial Perceived Quality
STD	Standard Deviation
SVC	Scalable Video Coding
SVCAT	Semantic Video Content Annotation Tool
TPQ	Temporal Perceived Quality
TE	Time Efficiency
UF	Utility Function
UCD	Universal Constraint Description
UED	Universal Environment Description
UMA	Universal Multimedia Access
UME	Universal Multimedia Experience

URI	Uniform Resource Identifier
URN	Uniform Resource Name
VQM	Video Quality Metric
WFL	Weber-Fechner Law
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
0-1 MCKP	0-1 Multiple-Choice Knapsack Problem

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Today's Web is characterized by very high heterogeneity in user preferences and available resources such as devices, terminals, network characteristics, and so forth. This makes accessing videos over the Web a challenging task. This heterogeneity in *usage constraints* (i.e., user and resource constraints) in the video access scenario can only be expected to intensify in the coming years. Indeed, users access videos with quite a variety of devices (e.g., PDAs, PCs, mobile phones, and so on) with different playback capabilities, including computational power, memory size, and display size. The network context in which accessing such videos takes place varies as well in terms of bandwidth, error rate, delay, jitters and reliability. At the same time, different formats coexist, including codecs, containers and metadata representation formats. Even more importantly, the users themselves are just as varied. They have different preferences in terms of video characteristics (such as preferred file formats, codecs, viewing size), personal tastes (e.g., interests in certain topics such as news), cultural background (such as religion and educational level), and requirements based on various profiles (e.g., age, country of residence, disabilities).

This tremendous variety in contexts poses challenges in targeting the concept of Universal Multimedia Access (UMA), which means that *'any multimedia content should be adapted to be transparently accessible at anytime, anywhere, and along with any devices, networks, and user preferences'* [16] [136]. To enable UMA, an intuitive solution would be to have multiple versions of a video content adapted in advance [78] [12]. This solution is inadequate in some cases as it requires more storage space, and the pre-adapted videos would only cover a subset of the specific contexts that are faced by users. The multimedia research community identified this problem several years ago and as a result has explored the possibility of dynamic adaptation to support UMA. Dynamic adaptation is defined as *'the process of adapting the video content to fit a usage environment or adapting the usage environment to accommodate the content'* [16]. Since the usage environment is usually inflexible and hard to change [22], researchers have focused on adapting video content on the fly according to the constraints of a specific context.

Initial efforts in the field of dynamic adaptation have targeted technical constraints raised by network limitations and the characteristics of the user's terminal. In order to satisfy these constraints, descriptions of the video content, so called metadata, are requisite to conducting the adaptation process. To be more precise, technical constraint-

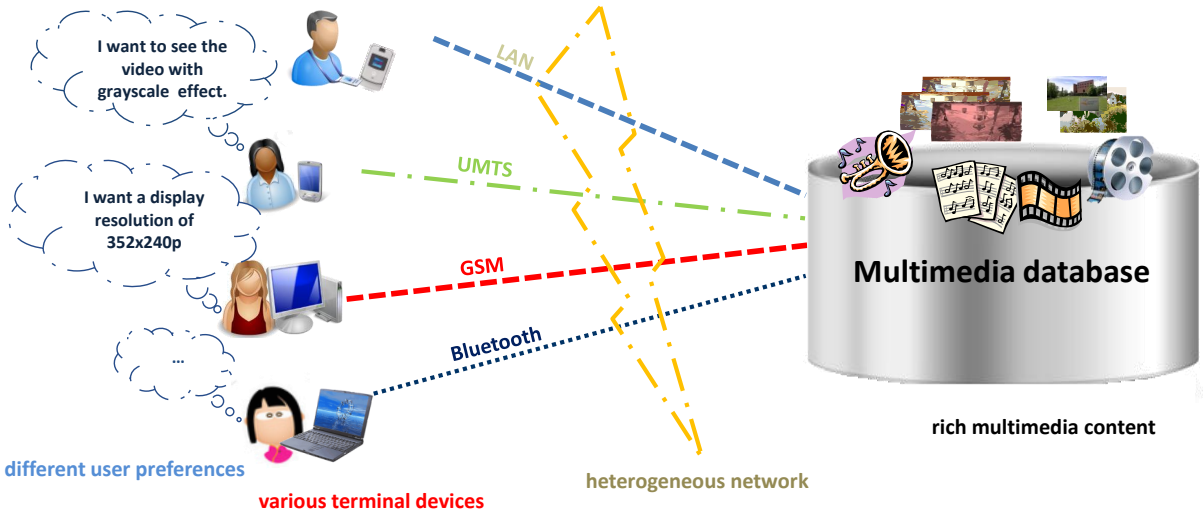


FIGURE 1.1 – Concept of Universal Multimedia Access (UMA) enables any users and devices to transparently access any multimedia content anytime and anywhere.

driven adaptation requires information on the technical metadata [105], including file creation, content modality (video, text, and so forth), format standard (such as AVI), format encoding (such as MPEG), frame size (640x480), frame rate (25 fps), and so on, as well as low-level features such as motion, color and texture. For instance, Figure 1.1 illustrates the case where description about the frame resolution of the video is needed, in order to assist the process of making adaptation decisions based on the various display resolution options of different types of consumption devices.

The research in this area is extensive, and a massive number of intelligent algorithms and adaptation approaches have been proposed. A survey of these approaches can be found in [69] [85]. Major examples of these approaches are transcoding (same content modality with format transformation such as temporal resolution reduction, e.g., reducing the frame rate from 30 fps to 10 fps), transmoding (same format with modality conversion such as video to slide show conversion), and scalable coding (keeping the same format and same modality such as scalable content adaptation). However, *'the final point in the multimedia consumption chain is the user and not the terminal'* [99]. Therefore, to put the user back in the center of the adaptation chain, a personalized, dynamic adaptation of the video content taking into account user preferences is crucial.

Before analyzing more deeply personalized dynamic adaptation approaches, it is worthwhile to have a good understanding of user preference terminology. So far, the term 'user' has been assigned to the person who accesses and consumes video content. In this thesis, the concept of user refers to more than just the final content consumer. Let us assume that the adaptation operates in the context of a video-on-demand system, as depicted in Figure 1.2. Thus, three types of users interfere in the process : the *end-users* who access and consume the video created by the *owner*, using the system managed by the *distributor*. Moreover, these users have different constraints, so called user preferences, regarding video content visualization, access, delivery and so on. In the following section, we analyze in turn end-user preferences, owner preferences, and distributor preferences.

End-user preferences are defined as *implicit* or *explicit* preferences. Implicit end-user preferences are inferred based on the information provided in the end-user profile

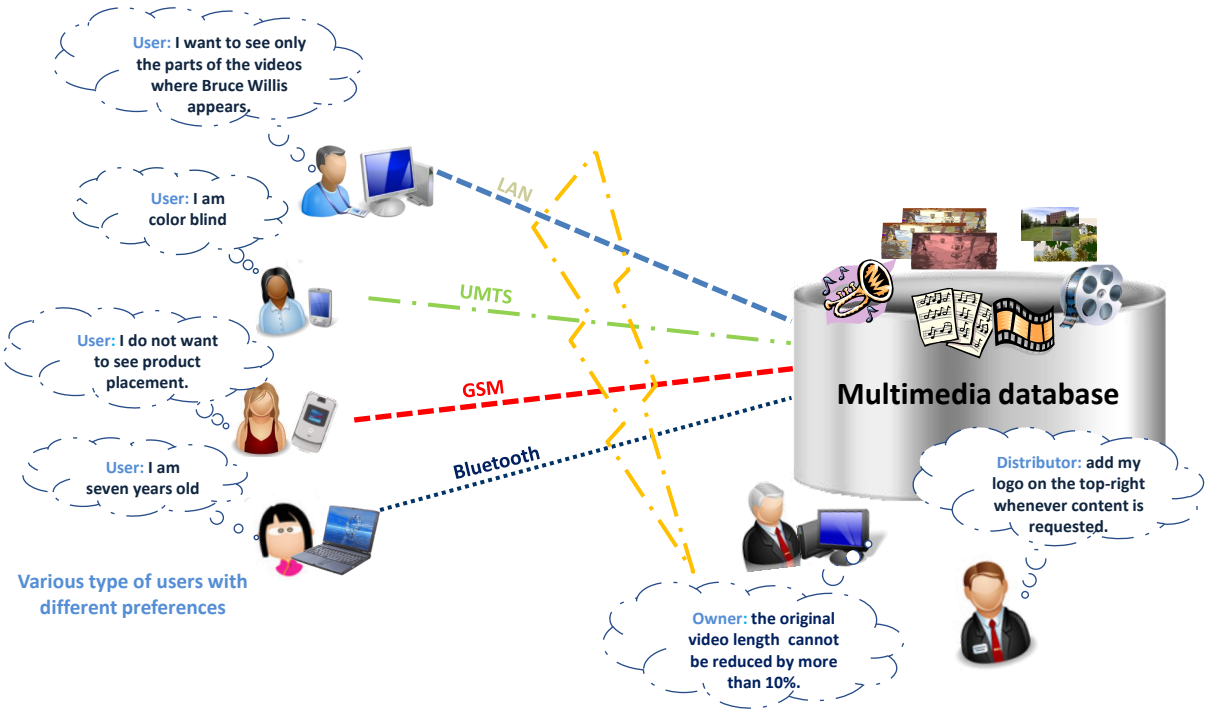


FIGURE 1.2 – Concept of Universal Multimedia Experience (UME) : any users should have an informative experience anytime, anywhere.

(e.g., 'She is seven years old' => she is not allowed to see violent scenes). Explicit end-user preferences are the preferences stated in her profile (e.g., 'I don't want to see product placement'). There are three different types of constraints related to these preferences : *technical constraints*, *perceptual constraints* and *semantic constraints* (see Figure 1.1). The *technical constraints* are expressed by the user in terms of video technical characteristics, as in, "I want a display resolution of 352*240p." As previously stated, information regarding the low-level features and technical metadata of the video is required to drive this personalized technical adaptation. The *perceptual constraints* express specific human perceptual preferences [85] related to perception limitations raised by the natural environment (e.g., if a user is too far from the screen, then the text font size may have to be increased), or visual impairments (e.g., I am color blind, so a color transformation is needed), etc. Perception-driven adaptation requires information regarding the perceptual arousal of the video [45] (e.g., intensity contrast, color scheme, etc.). The *semantic constraints* are constraints directly related to the semantics of the video content. They express the end-user's preferences in terms of personal taste (e.g., "I don't want to see product placement."), interest (e.g., "I want to see only the parts of the video where Bruce Willis appears."), etc. In order to perform personalized adaptation according to these semantic constraints, the structure and semantics of the video content should be described using semantic concepts [122] (e.g., human activity 'swimming', a particular object in a video scene 'actor', etc.).

On the other hand, the owner's preferences should also be taken into account. They involve constraints related to intellectual property rights over the content. These constraints are independent of the end-users. They convey explicit limitations on manipulating the video (e.g., the original video length cannot be reduced by more than 10 %). Indeed, international common ground on the legal definition of intellectual rights on media exists, since the majority of nations worldwide have signed the Bern Convention for the Protection of Literary and Artistic Works [112]. In particular, articles **9**

and **14** of the convention guarantee the right of the authors (i.e., owners) to refuse certain modifications of their works. Based on these legal considerations, any personalized adaptation that does not satisfy the *owner preferences* should be refused; otherwise, it is a violation of the owner's intellectual property rights.

Finally, as the manager of the multimedia database, the distributor is in charge of enforcing end-user and owner preferences whenever a video is accessed. Moreover, the distributor may also define his own constraints regarding content adaptation. For example, he could require the logo of his company to be added to the top right corner of the image for all distributed videos.

Many video manipulation techniques, so called adaptation operations, have been reported in the literature to have performed a personalized dynamic adaptation [15] [22] [12] [82]. The kind of manipulation performed depends strongly on the type of constraints expressed in the user preferences. In order to develop a coherent view toward these different solutions, several classifications of adaptation operations have been presented [20] [71]. These classifications include but are not limited to the following adaptation classes : format transcoding (i.e., changing the video coding format to make it compatible with the usage environment), scaling (i.e., producing alternative variations of the video through the selection or reduction of some elements of the video such as frames in a video clip or pixels in an image frame, to economize on resources), selection, removal and merging (i.e., changing the structure and the semantics of the video content to satisfy semantic constraints as in the selection of the scoring events in a football match, the filtering of violent scenes, and the removal of product placement), replacement (i.e., replacing selected elements in the video with other elements, as in, video sequences being replaced with representative visual images to create a slide show), and synthesis (i.e., analyzing the video content and presenting it in a new, synthesized form, as in presenting a hierarchical summary of a video by extracting the key frames and organizing them in a hierarchical way). Depending on the application scenarios, various combinations of these adaptation operations can be used. For an overview and detailed analysis of video adaptation, the reader is referred to the State of the Art Part I.

While existing adaptation operations cover most of the constraints mentioned previously, they do not guarantee providing the end-user with adapted content of good quality. This quality is multidimensional in nature. For instance, adaptation operations such as the removal of objects and the sequence of frames can produce undesired visual artifacts and temporal impairments in the video content, respectively. Thus, they affect the quality at the perceptual level. Furthermore, this removal may cause a loss of information that is important in understanding the video, and thus impacts the quality at the semantic level. These observations emphasize the need for maximizing the quality experienced by the user instead of simply enhancing the accessibility of the content as in most existing UMA systems. Indeed, empirical studies conducted in [26] reveal that a reliable system can completely fail in terms of user adoption due to the gap between service performance and user experience. To this end, the concept of Quality of Experience (QoE) has been introduced to fill this gap. Consequently, the multimedia research community shifted its focus from UMA towards Universal Multimedia Experience (UME), stating that an end-user should have an informative experience anytime, anywhere [99].

In the last few years, personalized dynamic adaptation targeting UME has been

an active research area. In the context of technical-driven adaptation, intensive research has been conducted to enhance the traditional view of Quality of Service (QoS) with a user-level defined Quality of Perception (QoP), which is part of the Quality of Experience (QoE) [42] [126]. Nevertheless, the problem of maximizing the quality of experience in the context of semantic adaptation was poorly tackled [20]. Semantic adaptation at the object level, so called *semantic object-based adaptation*, involving the removal adaptation class is noticeably still an open issue. This is due to the complexity of putting in place a fine-grained adaptation solution that requires joint consideration of several other closely related issues, including the analysis of video content, the understanding and modelling of user preferences, and the definition of quality models based on the concept of QoE.

In this thesis, we study the realization of ***semantic object-based video adaptation***. Our approach takes into consideration the set of ***semantic constraints*** imposed by or on the end-user, while ***preserving the intellectual property rights*** of the video owner and ***maximizing the quality experienced by the end-user***, thus enabling one more step toward the achievement of the UME vision.

1.2 Thesis Outline

The remainder of the thesis is organized as follows :

Chapter 2 - Motivation : motivates and illustrates our approach by presenting some scenarios benefiting from a semantic object-based adaptation. After analysing and discussing these scenarios, it derives their specific related research questions and outlines the contributions of the presented thesis.

Part I - State of the Art

Chapter 3 - Video Adaptation Background : provides the principal concepts and definitions about video content adaptation, and discusses the requirements to design an adaptation engine. It also gives an introduction to the basic concepts and objectives of the MPEG-7 and MPEG-21 standard with a special focus on the parts that are related to our work.

Chapter 4 - Related Work : assesses the most relevant research activities to the presented work that have been reported in the literature.

Part II - Formal Modelling of PIAF : Personalized vIdeo Adaptation Framework

Chapter 5 - General Adaptation Framework : presents our PIAF MPEG-based personalized video adaptation architecture and discusses the functionality and requirements of each module.

Chapter 6 - Preliminaries : reviews the fundamental properties of the concepts used throughout this thesis and introduces the formalisms that we have adopted for representing them.

Chapter 7 - Video Model : establishes a formal model for representing the spatial, temporal, and semantic information related to a video content.

Chapter 8 - Semantic Constraint Model : provides formal definitions of the end-user semantic constraint and the owner constraint. Moreover, it formally defines the information resulting from the semantic constraint instantiation process.

Chapter 9 - Metadata-driven Utility-based Adaptation Engine : describes our Utility Function (UF) model applied by PIAF's Adaptation Decision Taking Engine (ADTE) to support the generation of an adaptation plan for a given instantiated constraint. This function assesses the utility of a plan according to five parameters, with the aim of maximizing the quality experienced by the end-user. This chapter also presents the formal definitions of the five quality parameters used by the UF.

Chapter 10 - Adaptation Plan Computation : formally describes the computation of an adaptation plan with and without owner constraints. The former can lead to an optimization problem since satisfying the end-user constraints may violate the owner's intellectual property rights, thus resulting in conflicting constraints. The originality of this chapter is in formulating the adaptation plan computation problem as an optimization problem, and mapping it to the 0-1 Multiple-Choice Knapsack Problem (0-1 MCKP).

Part III - Implementation of PIAF

This part presents the implementation of the content description, context description and filtering modules within the PIAF architecture, based on the methodology and models described in previous part.

Chapter 11 - SVCAT : Semantic Video Content Adaptation Tool : presents our Semantic Video Content Annotation Tool (SVCAT), which assists annotators in generating video annotations compliant with the video model proposed in Chapter 7. Experimental evaluation regarding the accuracy and performance of SVCAT, as well as a positioning among the existing annotation tools, are also presented in this chapter.

Chapter 12 - Context Description and Filtering : implements the semantic constraint model presented in Chapter 8 using the MPEG-7 and MPEG-21 description tools. Moreover, an extension to the MPEG-7 and MPEG-21 standards is proposed to enable the representation of end-users semantic constraints.

Part IV - Evaluation of the Adaptation Decision Taking Engine

Chapter 13 - Utility Function Evaluation by Subjective Testing : describes the subjective assessment methods that we used to evaluate the per-

formance of the Utility Function (UF) model and its parameters. The performance is determined from a comparison between viewer ratings obtained in controlled subjective tests and quality predictions from the UF model. These experimental evaluations demonstrate that the UF model is a good predictor of the QoE reported by the users.

Chapter 14 - Conclusion and Future Works : concludes the thesis. After summarizing the major contributions, it discusses future extensions of this thesis.

Chapter 2

Motivation

2.1 Introduction

In this chapter, we motivate and illustrate our approach by presenting some scenarios in which users and applications in general may benefit from a semantic object-based adaptation. These scenarios are analyzed and discussed in Section 2.3 to identify the user requirements specifications. In Section 2.4, we list the requirements of an end-to-end video adaptation framework, examine and discuss each of them, derive their specific related research questions and present corresponding solutions that are the contributions of this thesis.

2.2 Application Scenarios

In the following, we present two application scenarios that illustrate the objectives of this thesis.

2.2.1 Application Scenario 1 : Personalization of Video content in Movie Download Website

Pascal is a member of the movie download website 'DownVid' [1]. Like a lot of people, Pascal considers product placement intolerable, as reported by a number of campaigns [2] that were launched particularly against embedding product information into media (e.g., movies). Indeed, this type of advertisement has grown exponentially in recent years, becoming the preferred alternative to traditional advertising. For instance, Brandchannel [3] provides a list of featured brands in each released film since 2001.

Pascal requests to watch 'Dark Tree' [4] movie without visually displayed brand names. A sample of featured brands within this movie are Absolute Vodka, Chevrolet,

1. DownVid is not a real website name.
2. http://www.commercialalert.org/
3. http://www.brandchannel.com
4. Dark Tree is not a real name of a movie. It is an excerpt from the film 'Sex and the City' that we will use in all our examples throughout this thesis.

Burger King, Honda and Sony Vaio [5]. To meet Pascal's demand, the video content needs to be adapted either by removing product placement objects in each frame or removing the whole video segments in which placed objects appear.

Furthermore, to guarantee the intellectual property of the video owner, assume that DownVid allows the owners to impose constraints forbidding some modifications that could damage their video. For instance, the owner of 'Dark Tree' can impose a restriction on the number of frames that may be deleted that is, the video cannot be shortened more than 10%. Furthermore, the owner can forbid the removal of some product placements such that Chevrolet cars for business reason. Therefore, the adaptation decision should be made such that the property rights of the owner and the end-user's constraints are satisfied at once.

Besides the owner and end-user constraints, the content distributor (i.e., DownVid) can also enforce some level of control as to how the video content is actually accessed, by inferring restrictions from the end-user profile. Indeed, according to the information in his profile, Pascal is a Turkish citizen. In Turkey's legislation, alcohol and tobacco advertisements are banned, otherwise a fine has to be paid. For instance, a concrete incident is a Turkish private television channel, which paid a fine of 50,000 liras ($33,000) for broadcasting the classic Tintin cartoon series featuring smoking scenes [6]. Therefore, to prevent the violation of the local laws, 'Absolute Vodka' has to be removed from the video content before this latter is sent to the end-user.

2.2.2 Application Scenario 2 : Privacy-Preserving Video Sharing in Social Networks

Martin and Getnet are invited to Anna's birthday party. Martin is an amateur videographer and an active member on Google+ [7]. At each event he attends, he creates a blog with the videos that he captured, comments and shares them with his friends. Tonight, he will record Anna's birthday party. A few days after the party, Martin shows the video to his two friends and asks about their opinion before posting it on his Google+ account. Anna is so happy to see the video and asks if she can share it with her friends. However, Getnet does not like being in a video that will be seen by people he doesn't know. In such a case, a simple grant or denial of access to the video content is not enough. Instead of editing the video and storing multiple versions of the same video for different friends profile, Martin simply annotates the video parts where Getnet is shown, and configures the privacy settings based on the end-user profile. Upon request, Martin, Getnet and their common friends (including Anna) will be able to watch the original video whereas the others will see the adapted one without Getnet being in it.

5. We apologize for this implicit advertisement
6. http://www.itnsource.com/shotlist/RTV/2010/02/19/RTV436710/?v=2
7. http://www.google.com/+/

2.3 Application Scenarios Analysis and User Requirement Specifications

The application scenarios listed above clearly outline the importance of having a semantic object-based adaptation, which can personalize the video content according to semantic constraints defined by the users. For instance, the first scenario requires the removal of the product placement objects, and the second scenario requires the removal of the person 'Getnet'. In the following, we analyze both scenarios in order to delineate the user requirement specifications for realizing video adaptation in this type of application.

a. End-User Requirements

- **Performing a personalized adaptation of the video content according to object-based semantic constraints :** back to scenario 1, DownVid must dispose of a removal adaptation technique so that Pascal can watch the video without product placements. Likewise, for the end-user Getnet in scenario 2, who requires being removed from the video.

- **Providing an automatic transparent personalized adaptation :** the system should automatically map the end-user preferences on the video content, locate the parts to be adapted and perform the adaptation without requiring any intervention from the end-user.

- **Realizing a quality of experience-aware adaptation :** given a specific semantic constraint, several possible adaptation techniques can be executed over the video content. For instance, to satisfy Pascal's constraint, we can remove the product placement objects by blurring them, or simply remove the parts of the video containing these objects. Each of these solutions produces an adapted video that has a different impact on the overall end-user's satisfaction. Therefore, the adaptation process must be able to predict the impact of each adaptation option in order to provide the end-user with the best Quality of Experience (QoE). To this end, several factors must be considered for each scenario to maximize the QoE of the end-user. In the following, we list three factors and explain them by simply referring to one scenario at once :

  - *Semantic quality of the adapted content :* in scenario 2, imagine Getnet appearing in the video part where Anna is cutting her birthday cake. Removing this segment would drastically affect the semantic quality of the adapted video.

  - *Perceptual quality of the adapted content :* in scenario 1, one can remove the product placement objects using an inpainting technique ; that is, replacing the pixels of the unwanted object with the neighbouring pixels. However, if the object is large, the inpainting process can cause noticeable visual artifacts, resulting in an adapted video of very low perceptual quality.

  - *Time efficiency of the adaptation :* in scenario 2, removing Getnet in each frame using an inpainting technique can provide a better semantic and perceptual quality than removing the whole segments. However, the execution

time of the second option could be much lower than the first option, thus resulting in a much better quality of experience for the end-user.

b. Owner Requirement

Protecting owner intellectual property rights : legally, the owner has the ultimate decision-making-power over the modification of his video. Therefore, the application should always grant him a way to specify adaptation constraints, and enforce them with a higher level of priority. For instance, in scenario 1, it is not possible to remove some product placement because the video owner has explicitly forbidden it.

c. Distributor Requirement

As previously stated, the distributor is responsible for enforcing the end-user and the owner constraints. In particular, he must infer some information from the end-user profile to prevent the violation of legislation as shown in scenario 1. In this thesis, we restrict ourselves to processing the end-user and owner preferences. Indeed, the constraints of the distributor are either similar to the constraints of the owner or of the end-user.

2.4 Requirements of Video Adaptation

In this section, important issues pertaining to video adaptation are discussed. To begin with, we provide a general description of the structural and semantic video elements, thus recalling the readers with the concepts related to the video. Then, we discuss in detail the specifications of an end-to-end video adaptation framework.

2.4.1 Representation of the Video Elements

"Video data is naturally complex; therefore a thorough understanding of its unique characteristics is essential to develop techniques for managing it," according to Tjondronegoro [128]. Approving this statement, we present in this section the essential information about the video, which is important for the understanding of the adaptation problem and its related challenges. Figure 2.1 illustrates an example of the representation of video elements.

A video can be seen as a sequence of image frames, which convey a rich semantic presentation through synchronized audio, visual and text presentations over a period of time [128]. Thus, the fundamental units of a video are single image frames [122] as they are atomic on the time axis [37]. Moreover, *"a key characteristic of video data is its associated spatial and temporal information that delivers semantically coherent narrative,"* according to an article by Ren et al. [108]. This information is mainly inherited from the image frame itself due to the spatial and semantic relationship between its elements [62]. Back to Figure 2.1, a red circle (local structure described with two low-level features : color and shape) associated with semantics (i.e., an apple-

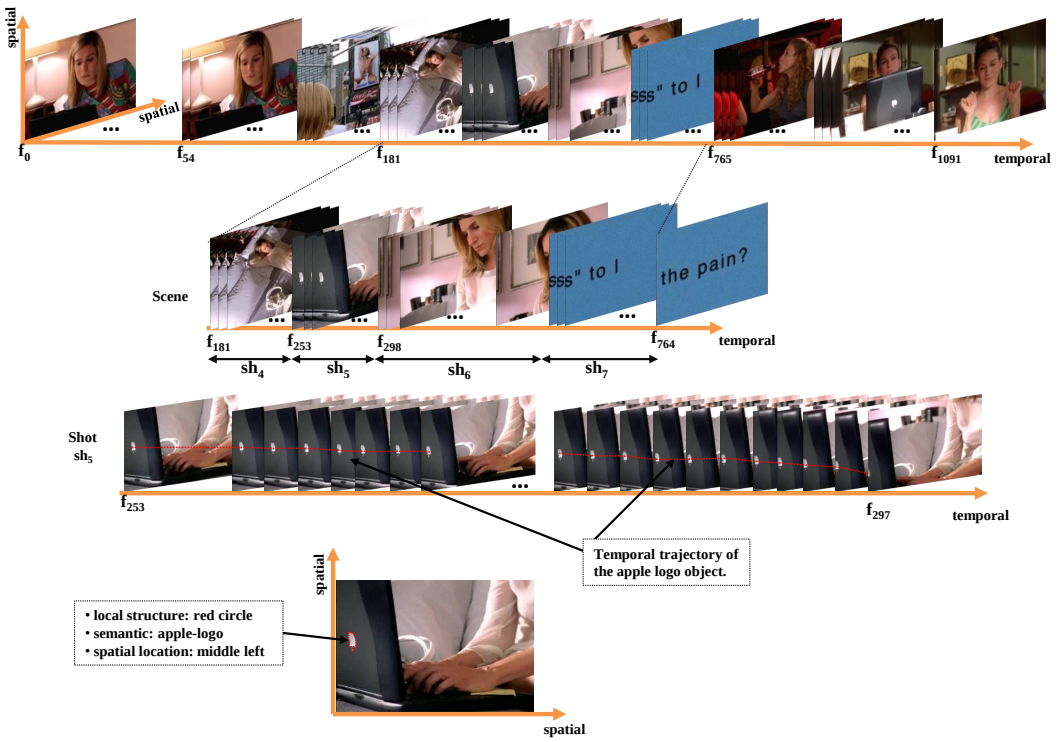


FIGURE 2.1 – Representation of the Video Elements.

logo) becomes an object (i.e., an apple-logo object), which is spatially located at a specific position in the image frame (e.g., middle left), and can be related spatially to another object in the same image frame (e.g., on the back of the laptop). In addition, due to the temporal relations between consecutive image frames, the information about the relations between these objects and their motions are propagated over time creating a so called temporal trajectory [108]. These temporal trajectories of spatial relations among objects as well as temporal object trajectories are important as they reveal the semantic evolution of spatial properties over time [108], thereby creating semantic temporal sequences of image frames (e.g., scoring event in football). In the field of multimedia analysis, the video is defined as a sequence of scenes, whereas each scene consists of several sequences of frames that are semantically related and narrating the same events [139]. These sequences of frames, so called shots, are defined as *"one uninterrupted image captured with a single static or mobile framing"* [14] from the same camera direction and view angle.

2.4.2 Specifications of an end-to-end Video Adaptation Framework

As stated in Chapter 1, adaptation is required to cope with the challenges imposed by the UMA paradigm, and enable the vision of UME. In the adaptation framework, the adaptation engine is the technical realization of the adapting functionality, which transforms the video content from its initial state to a final state in order to satisfy the constraints related to the usage environments and users preferences. In the following, specifications of an end-to-end video adaptation framework are discussed.

a. Context Awareness

The description of the context is a mandatory step in video content adaptation framework. To effectively adapt the video content, the adaptation engine should have exact information about the usage constraints (network, device, terminal, users, etc.) in order to decide *what* granularity level of the video (e.g., pixel, object, frame, shot, etc.) should undergo *which* adaptation techniques(e.g., frame rate reduction, object removal, etc.). Back to application scenario 1, the adaptation engine should be aware of the user constraints in order to remove the product placements from the 'Dark Tree' movie according to Pascal's needs while preserving the intellectual property rights of the movie owner and considering the DownVid constraints. To avoid interoperability problems of the information exchange, several international organizations such as the World Wide Web Consortium (W3C) [96] and Moving Picture Expert Group (MPEG) [16] have developed tools and protocols to describe the usage environment in a standardized way. For instance, Part 7 of MPEG-21, referred to as Digital Item Adaptation (DIA) [16], provides the Universal Environment Description (UED) tools to describe the user preferences, terminal capabilities, network conditions and the natural environment. Likewise, W3C defines the Composite Capability/Preference Profiles (CC/PP) protocol for context description.

b. Content Awareness

In order to accurately satisfy the usage environment constraints, the adaptation engine should be aware of the video content to some extent. Depending on the application scenario, different types of constraints require different types of descriptions to drive the adaptation process [131] [105]. For instance, technical constraints-driven adaptation requires information about the content (e.g., file creation, format encoding (MPEG), frame size (640×480), etc.), and information presented in the content (i.e., low-level features). Whereas, semantic-driven adaptation mainly requires information presented in the video content (i.e., semantic and structural level). These descriptions may come from both manual annotations and automatic content analysis techniques. If the descriptions are pre-computed, the adaptation engine makes use of it to automatically decide on the adaptation to perform. If the descriptions are to be extracted on the fly, the adaptation engine should be able to dynamically analyse the content before taking the adaptation decision. Recognizing the importance of the video descriptions in the adaptation framework, several standardized tools have been developed targeting different applications. In turn, MPEG-7 has standardized a comprehensive set of description tools, that are, Descriptors (D)s and Description Schemes (DS)s to describe the information present in the content [52] [87].

In the context of personalized semantic adaptation, the availability of standardized semantic descriptions for both video content and user constraints, highly contributes to the automation of the adaptation process. Indeed, the availability of these descriptions allows intelligent interaction over the content such as automatic content filtering according to user preferences. This filtering provides the information needed to drive an automatic adaptation process at a later stage. Thereby, the adaptation is transparently done for the users giving them the impression that the interaction is only with the content.

c. Quality of Experience Awareness

A vital prerequisite to enable the vision of UME is to have a QoE-aware adaptation engine [99]. In the literature, the term *"Quality of Experience"* has been given several definitions depending on the application perspective (e.g., from a business perspective, QoE is a subjective measure of a customer's experiences with a vendor). An overview of formal and informal definitions of QoE can be found in [11] and [145]. In the video adaptation context, QoE represents the degree of satisfaction of the end-user with respect to an adapted video. MPEG-21 DIA [16] has presented the concept of *utility* as a measurement of the QoE resulting from an adaptation operation. Utility can be measured at three different levels [139] : the objective level (e.g., Video Quality Metric (VQM) [100] and Peak Signal-to-Noise Ratio (PSNR) [9]) ; the subjective level (e.g., the end-user gives subjective scores) ; and the comprehension/semantic level (e.g., measuring the amount of information assimilated by the end-user). Several video quality metrics were presented in the literature to measure this utility. An overview of these metrics can be found in [141] and [92]. Given the complex nature of utility, it is difficult to come up with a universal quality metric for different levels. In practice, the quality metrics are modelled with regards to specific application scenarios. For video adaptation, a metric called Utility Function (UF) has been defined by Wang *et al.* [138] to measure the adaptation utility for UMA, and later standardized in MPEG-21 DIA [16]. UF describes the trade-off relationship between constraints and utilities along each adaptation dimension. It plays a key role in choosing the optimal adaptation among multiple options that meet the usage environment constraints.

2.5 Problem Statement

In this thesis, we aim to contribute to the Semantic Protection and Personalization of Video Content by considering the set of high-level semantic constraints imposed by or on the end-user while preserving the intellectual property rights of the video owner and maximizing the quality perceived by the end-user after adaptation. Based on the requirements expressed in the previous section, we formulate the semantic video adaptation problem as follows :

Given *video content* and *object-based semantic constraints* specified by *different types of users*, develop efficient adaptation decision techniques to identify the *best spatial-temporal adaptation plan*, which *maximizes the global quality* of the adapted content while *respecting the owner constraints*.

To resolve this problem, a number of research issues must be addressed, which are related to analysis of video content, understanding and modelling of user preferences, and definition of quality model based on the concept of QoE.

2.5.1 Research Problems

To facilitate the achievement of our objective, the task involves research efforts from several aspects :

- **Research Problem 1 : Generating standardized content and context descriptions at the object-level ;** despite the existence of many annotation tools compliant to a standard (see survey[25]), these tools have a number of limitations with respect to the annotation of objects. In particular, they cannot provide accurate selection of the local structure of the object and automatically locate it in each frame in which it appears. Moreover, to the best of our knowledge, there is no standard that enables the description of the semantic preferences of end-users at the object-level.

- **Research Problem 2 : Objectively measuring the quality of the adapted video ;** many adaptation operations can satisfy the same semantic constraint, thus resulting in several possible adaptation plans to be executed over the video. However, the resulting adapted videos will have different levels of quality. Thus, in order to identify the best adaptation plan, a quality metric for objectively measuring the quality of the adapted video according to the multiple aspects of the QoE is needed. Several objective video quality metrics already exist in the literature [100], [9] and [92]. These metrics are mainly modelled for application scenarios involving technical/perceptual constraints-driven adaptation, and do not consider all the aspects of the QoE. For instance, the traditional video quality metrics such as Mean Squared Error (MSE) [49] and PSNR [9], though computationally simple, are known to disregard the characteristics of human visual perception [140]. They always operate on the whole frame and do not consider any other important factors (e.g., Human Visual System (HVS) characteristics) that can strongly influence the perceived quality. Moreover, these fidelity metrics are not suitable for measuring the adaptation quality in the context of object removal, since they aim to predict the visibility of image reproduction errors based on the original un-deteriorated images, which are not available in this situation. Finally, these metrics are unable to measure the amount of information assimilated by the end-user after an object-based adaptation, since they are not developed for application scenarios involving semantic constraints-driven adaptation.

- **Research Problem 3 : Resolving Conflicting Constraints ;** taking into account the owner constraints in addition to the end-user constraints in the adaptation process leads to the possibility of conflicting constraints. In fact, resolving some owner constraints such as 'specifying a percentage of the number of the frames that can be dropped' amounts to an optimization problem. To this end, we need a methodology able to resolve these conflicting constraints while preserving the owner's intellectual property rights and maximizing the QoE of the end-user.

2.6 Summary of Contributions

The main contributions of our work in solving the aforementioned problems are summarized as follows :

2.6.1 Methodological contribution

- **UF model** : we define a utility-based model to support the generation of a plan in order to adapt a video to an end-user constraint. Accordingly, we propose a Utility Function (UF), which integrates semantic concerns with user perceptual consideration, to evaluate the effect of adaptation operations. This function assesses the utility according to five parameters with a threefold purpose : 1) preserving the semantic integrity of the content by minimizing the overall impact of the adaptation especially on semantically critical parts of the content ; 2) maximizing the spatial and temporal perceived quality of the adapted content ; and 3) minimizing the processing cost of the adaptation operation. Experimental evaluation based on a user study demonstrates that the UF model is a good predictor of the QoE reported by the users : the predicted ratings show a strong correlation of 0.84 with the Mean Opinion Score (MOS) ratings. This contribution is detailed in Chapter 9 and Chapter 13.

- **Formalization of the end-user semantic constraint :** we formalize these constraints and propose an extension to the MPEG-7 and MPEG-21 standards to include them under a new type of preferences called *UsageSemanticPreferences*. This contribution is detailed in Chapter 8 and Chapter 12.

2.6.2 Theoretical contributions

- **Resolution of conflicting end-user and owner semantic constraints ;** for the case of conflicting constraints, we formulate the optimal adaptation plan computation problem as an optimization problem. The originality of this contribution is in mapping the adaptation plan computation problem to the 0-1 Multiple-Choice Knapsack Problem (0-1 MCKP). This contribution is detailed in Chapter 10.

2.6.3 Software contributions

- **Object-based video content model and a semantic video content adaptation tool SVCAT compliant to the standard MPEG-7 :** we propose a content model for representing the spatial, temporal, and semantic information related to a video. The originality of this model lies in particular in its expressive representation of the spatial, temporal and semantic properties of video objects. Moreover, we develop a Semantic Video Content Annotation Tool (SVCAT) that assists annotators in generating video annotations according to the proposed video model. The novelty of SVCAT lies in its automatic propagation of the object

localization and description metadata realized by tracking their contour through the video, thus drastically alleviating the task of the annotators. Experimental results show that SVCAT provides accurate metadata to object-based applications with nearly exact contours of multiple deformable objects. This contribution is detailed in Chapter 7 and Chapter 11.

– **Personalized vIdeo Adaptation Framework (PIAF) :** all the developed contributions were carried out as part of a coherent framework called PIAF. PIAF is a complete modular MPEG standard compliant framework that covers the whole process of semantic video adaptation. This contribution is detailed in Part III.

Part I

State of the Art

Chapter 3

Video Adaptation Background

The primary objective of this chapter is to familiarize the reader with the principal concepts and definitions about video content adaptation. To begin with, we explain the need for content adaptation and discuss the requirements to design an adaptation engine. We mainly present a classification of the adaptation approaches, and overview the existing adaptation decision-taking methods. Moreover, we discuss the quality aspects in the context of video adaptation that are, the semantic quality and perceptual quality. Finally, since the content and context descriptions are mandatory to drive an adaptation, we conclude this chapter by an overview of the MPEG-7 and MPEG-21 standards with an emphasize on the parts that are related to the work of this thesis.

3.1 Video Adaptation Engines in Universal Multimedia Access (UMA)

As discussed in Chapter 1, video adaptation is the key technique to ensure UMA, which refers to ubiquitous access to and convenient consumption of :

- **any multimedia content** independently of
  - *the modality* : image, audio, video, text, image2D, image3D, etc.
  - *the container format* : GIF/TIFF/FITS for image, AVI/IIF for video, etc.
  - *the encoding format* : JPEG/PNG for image, MPEG for video, etc.
- **from any devices :** PDA, PC, Mobile phone, etc.
  - *with different capabilities* : computational power, memory size, display size, etc.
- **anytime across networks** of different characteristics including :
  - *static capabilities* such as its maximum capacity, etc.
  - *dynamic conditions* such as the available bandwidth, error, delay, etc.
- **anywhere** according to the usage environment conditions such as :
  - *terminal capabilities* : encoding and decoding capabilities, display and audio output capabilities, power, storage, and input/output characteristics of a device.
  - *natural environment* that pertains to physical environmental conditions such as the lighting conditions, auditory noise level, or a circumstance such as the time and location that content is consumed or processed.
- **satisfying any type of constraints** defined in the end-user preferences :
  - *technical constraints* raised by network limitations and the characteristics of the end-user's terminal.

  - *perceptual constraints* related to perception limitations raised by the natural environment, visual impairments, audio impairments, etc.
  - *semantic constraints* related to user personal taste, interests, mood, etc.
- **while preserving restrictions** imposed by :
  - *intellectual property rights of the owner* that convey explicit limitations on manipulating the video.
  - *provider constraints* like i.e., add the logo of my company on the top-right of the data content.

Given this large variety in contexts, different type of users with different type of constraints over heterogeneous network accessing multimedia content through various devices, it is impossible to have an adaptation engine that copes with the full spectrum of possible requests and constraints, especially as the world of multimedia keeps evolving [71]. To this end, the design of an adaptation engine is strongly dependent on the requirements of the UMA application. Indeed, six questions must be answered during the designing stage : Q1) what is the granularity of adaptation ? ; Q2) what is the type of adaptation technique to be used ? ; Q3) what is the target of the adaptation ? ; Q4) how the adaptation plan is generated by the Adaptation Decision Taking Engine (ADTE) ? ; Q5) when does the adaptation take place (e.g., online, offline, etc.) ? ; Q6) and where does the adaptation take place (e.g., servers, clients, proxies, etc.) ? In the remainder of this chapter, we provide the relevant basics to answer the first four questions, which are related to this thesis. In particular, we present a classification of the adaptation approaches and discuss the existing adaptation decision-taking methods. Moreover, we discuss both terms semantic quality and perceptual quality, which are key aspects of the QoE.

3.1.1 Granularity Level of Adaptation

A granularity level of adaptation is the *'what to adapt'* in an adaptation framework. It refers to the video elements (e.g., pixel, object, frame, shot, etc.) that should be adapted. As previously discussed in Section 2.4.1, the video elements can be found at the spatial dimension (e.g., pixels and objects contained in the frame), temporal dimension (e.g., scenes, shots, frame sequences) and spatio-temporal dimension (e.g., frame sequences related to a specific object).

3.1.2 Types of Adaptation Operation

An adaptation operation is the manipulation technique that is executed on a video element. As stated in Chapter 1, depending on the type of constraints (i.e., technical constraints, perceptual constraints and semantic constraints) expressed in the user preferences, different type of adaptation operations can be used. Based on the classification of adaptation operations presented in [20] [71], we distinguish between six types of adaptation operations : 1) format transcoding ; 2) scaling ; 3) selection ; 4) removal or reduction ; 5) replacement or substitution ; and 6) Synthesis. These adaptation are temporal, spatial, or spatio-temporal according to the granularity level on which they are executed. Some examples of these adaptation operations are depicted in Figure 3.1 (reprinted from [22], by W.H. Cheng).

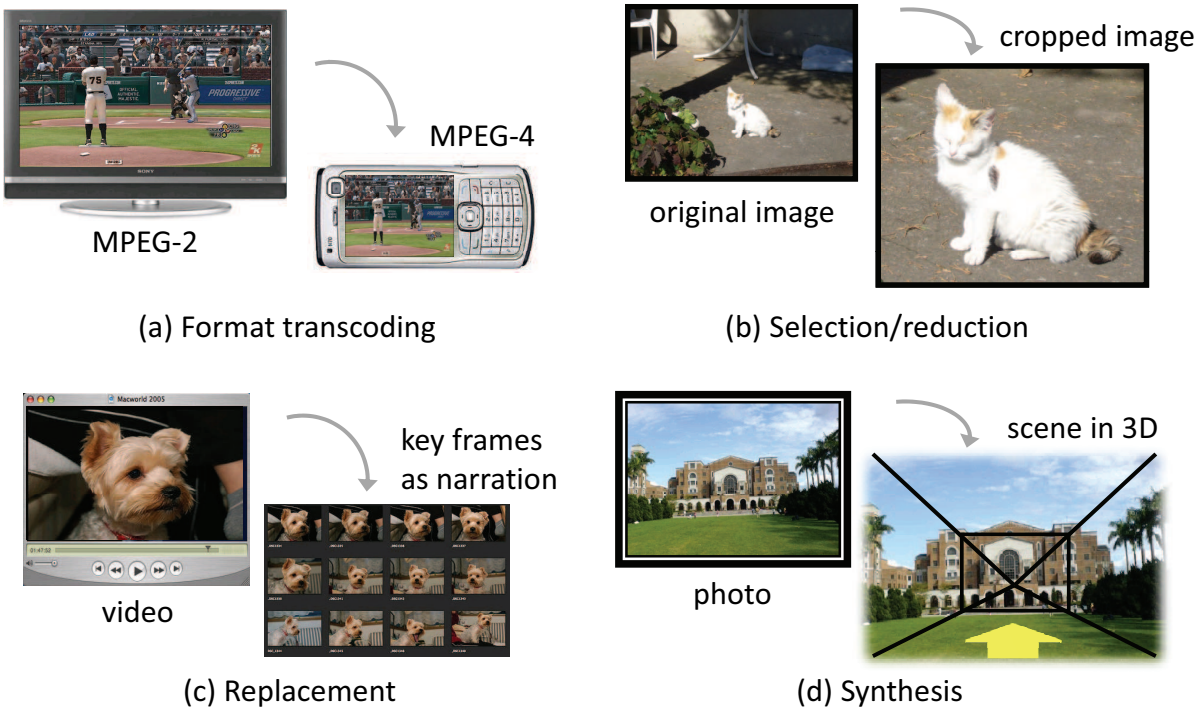


FIGURE 3.1 – Examples of some adaptation operations.

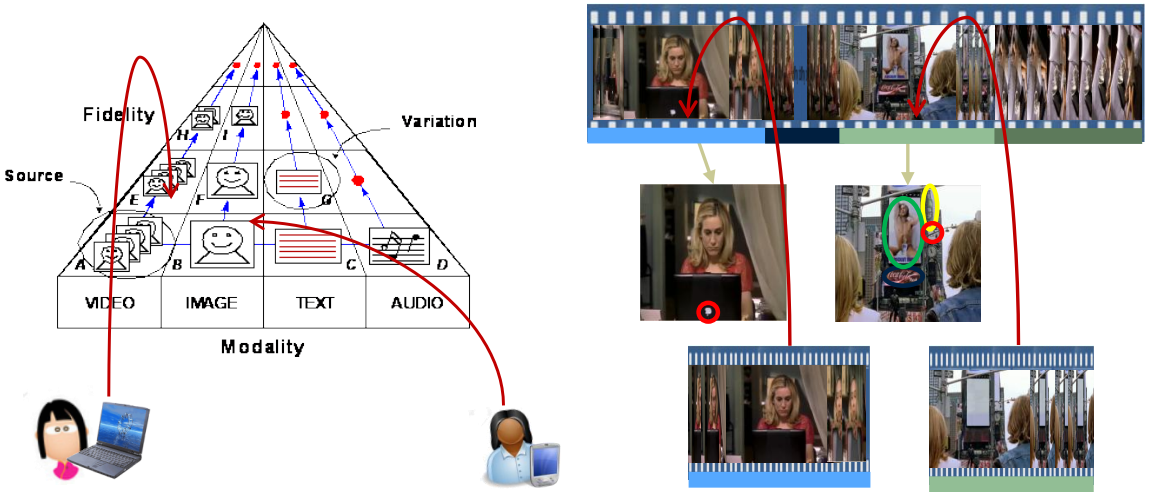
3.1.3 Classification of Adaptation Approaches

Several video adaptation approaches have already been proposed in the literature. To analyze this domain, we adopt the classification presented in this survey [85]. We distinguish two major categories of approaches : static adaptation and dynamic adaptation. Both approaches are detailed in the remainder of this section.

3.1.3.1 Static Adaptation Approaches

Static adaptation approaches assume the availability of several versions/variations of the same video content or even several alternative content parts, which are adapted in advance to address different kinds of usage constraints. Static adaptation is performed either by selecting the most adequate version of the content (see Figure 3.2a), or by substituting a part of content by a pre-adapted one according to the usage constraints (see Figure 3.2b). For instance, Figure 3.2a illustrates the adaptation by selection approach adopted in the InfoPyramid framework [78]. It depicts the source video in the lower left corner (A), and eight variations with different modalities : two variations are video programs (E, H), three are images (B, F, I), two are texts C, G), and one is audio (D). Each of the variation programs has a specified fidelity value that indicates the fidelity of the variation program with respect to the source video.

Static adaptation approaches are advantageous for application scenarios that require adaptation to specific kind of usage constraints with a reduced runtime processing during the end-user request. Moreover, static adaptation preserves the intellectual property rights of the owner since he/she has the full control on how the content should be adapted and delivered to the end-user.



(a) Adaptation by selection in the InfoPyramid framework [78] (reprinted from [52]).

(b) Adaptation by substitution in a product placement removal application.

FIGURE 3.2 – Static adaptation approaches.

However, these approaches have a number of disadvantages, mainly related to the management and maintenance of the different variations of the same content. Static adaptation approaches do not scale with the number of considered constraints. Whenever a new constraint is introduced, new content variations should be created for each existing video in the database. Consequently, each additional constraint multiplies by a factor the number of variations for a video. Therefore, in the case of a database containing a large number of videos, the required amount of storage space quickly explodes. Even if the storage space is not an issue, efficient strategies for organizing and retrieving the adequate variation for the end-user should be developed. Finally, static adaptation approaches assume that all adaptation constraints are known in advance. If a new constraint appears when the system is operating, it could only be dealt with by halting the system, recomputing new variations of the content (and as mentioned above, the number of new variations may be extremely high) and then resuming the operation. Clearly, this is not acceptable in dynamic environments, in which the videos should rather be adapted on the fly.

3.1.3.2 Dynamic Adaptation Approaches

Dynamic adaptation approaches are performed by transforming on the fly the video content from its initial state to a final state in order to satisfy a set of usage constraints. The adaptation solutions depend on the type of constraints that must be satisfied in a given application. We classify these adaptation solutions into three distinct categories : perception-driven, technical-driven and semantic-driven. This classification is illustrated with examples in Figure 3.3. In the remainder of this section, we overview each of these categories with a more focus on semantic-driven adaptation techniques, as being related to the scope of this thesis.

Perception-driven Adaptation Approaches ; they deal with the perceptual constraints, which express specific human perceptual preferences related to perception limitations raised by the natural environment, visual impairments, audio impairments, etc. [85]. This type of approach requires information regarding the perceptual arousal of the

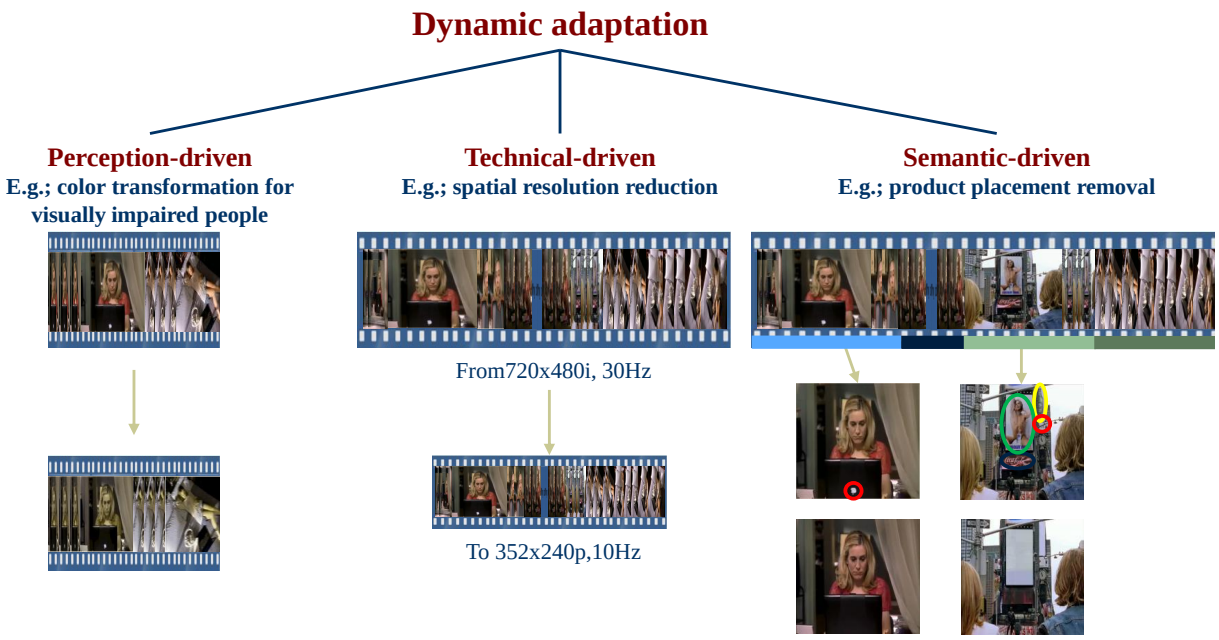


FIGURE 3.3 – Classification of the adaptation solutions.

video [45] [125] (e.g., intensity contrast, color scheme, emotion etc.), in order to be performed. Examples of adaptation approaches according to the natural environments include audio adaptation [35] (e.g., if a user is listening to the music in a crowded place, then the volume may have to be increased) or text adaptation (e.g., if a user is too far from the screen, then the text font size may have to be increased). With respect to visual impairments, the two most common categories are color vision deficiency (e.g., dichromacy or anomalous trichromacy), and low-vision impairments (e.g., light sensitivity where the user is extremely sensitive to the normal light condition). Thereby, in order to adapt for visually impaired people, information about the color, contrast, brightness, etc. is needed. A detailed description of these two categories is given in [94].

Technical-driven Adaptation Approaches ; they deal with technical constraints raised by network limitations and the characteristics of the end-user's terminal. These approaches require information regarding the low-level features and technical metadata of the video in order to be performed. In the literature, the most used techniques for technical-driven adaptation approaches are transcoding, transmoding and Scalable Video Coding (SVC). A survey of these solutions can be found in [69] [85] [56] [113] :

- **Transcoding :** is intended to decrease the required content resources and thus matching the available network/terminal consumption capabilities, keeping the same content modality [85]. This adaptation technique involves syntax/format conversion [3] (e.g., converting the format from MPEG-4 to MPEG-2 because the receiving device cannot handle the MPEG-4 coding method), temporal resolution reduction (e.g., reducing the frame rate from 30 fps to 10 fps), spatial resolution reduction [27] (e.g., changing the frame size from 720×480 to 352×240), bit-rate adjustment (e.g., changing the bit plane depth or color depth), etc. Transcoding adaptation requires signal-processing techniques that may be computationally intensive. Thus, it may be quite expensive to achieve a large scale adaptation deployment when many on-the-fly adaptations are required [85].

- **Transmoding :** refers to modality conversion while the original format is kept[69].

This adaptation technique is used when the usage environment do not allow the consumption of the content with its original modality. Transmoding techniques include, but are not limited to the following : video to slide show, video to text, video to image and video to audio.

- **Scalable Video Coding (SVC) :** is an extension of the H.264/MPEG-4 AVC video compression standard for video encoding [56]. As defined in [113], 'SVC enables the transmission and decoding of partial bit streams to provide video services with lower temporal or spatial resolutions or reduced fidelity while retaining a reconstruction quality that is high relative to the rate of the partial bit streams'. There are three main types of scalability [56] : 1) Spatial scalability means that it should be possible to decode the input video at a lower spatial resolutions (i.e., picture size reduction), 2) temporal scalability means that frames can be dropped in the bit streams (i.e., not all encoded frames will be decoded), thus resulting in a lower frame rate of the decoded video sequence (i.e., frame rate reduction), and 3) Quality scalability (also referred to as fidelity or Signal-to-Noise Ratio (SNR) scalability), means that the same spatio-temporal resolution of the original bit stream is kept, but with a reduction in the bit rate. This latter reduction results in a decrease of the visual quality. In many applications, these different types of scalability can be combined. An example of scalable coding is the scalable content with a bitstream syntax description presented in MPEG-21 DIA [16].

Semantic-driven Adaptation Approaches ; they deal with semantic constraints, which are expressed in terms of concepts/keywords given by the user. These approaches mainly involve the selection and reduction adaptation operations along spatial, temporal and spatio-temporal dimensions. Thus, information related to the structure and semantics of the video content is essential to perform any type of semantic adaptation. In the following, we overview some existing spatial and temporal semantic approaches :

- **Spatial Semantic Adaptation Approach :** can be achieved in several ways : 1) along the Region Of Interest (ROI) scalability axes by attaching a higher priority to the ROIs, so that only the quality of the none prior regions is reduced when a transcoding adaptation operation occurs [15] ; 2) through cropping and scaling the content by selecting a region that has a semantic meaning within the video, and its priority has been assigned beforehand [110], [129] ; 3) through spatial/scene summarization by removing unwanted objects or region from the frames based on user preferences [32]. Figure 3.4 illustrates an example of spatial semantic adaptation of a soccer game that involves cropping and scaling. For instance, assume that a user wants to watch a soccer game on his/her mobile phone. A simple spatial transcoding is not enough since it causes excessive spatial resolution reduction, where the ball will be hardly recognized. In order to improve the visibility of the users, a possible solution is crop the frames while preserving the ROIs (e.g., the soccer ball, player holding it, etc.), which should be defined beforehand.

- **Temporal Semantic Adaptation Approach :** can be essentially done in two ways : 1) temporal summarization by creating an audiovisual summary for a specific user [91] ; 2) temporal semantic reduction by dropping audiovisual unsuitable temporal segments (shots, events) based on user preferences [134] [149] :
  - *Temporal summarization* is an adaptation by synthesis, which aims to provide

the user with a more comprehensive experience or more efficient tool for navigation [20]. To do this adaptation, the content is first temporally segmented into hierarchical structures (e.g., scenes, shots, events, key frames) and then semantically annotated. Upon request, temporal segments are selected to create the summary that most satisfies the user preferences, e.g., summary of goal events in a soccer game. A survey of the existing summarization techniques is presented in [91].

- *Temporal semantic reduction* is an adaptation by removal or reduction. It depends strongly on the richness and precision of the video content description. To this end, the video should be first segmented into individual temporal segments, each of them with a semantic description. Upon request, the unwanted temporal segment are removed from the video to satisfy the user preferences. In case temporal semantic reduction is combined with transcoding as in [69], then the adaptation involves the removal of unwanted segment and/or spatial resolution reduction for segment with low priority for the user.

– **Spatio-temporal Semantic Adaptation Approach :** combines both spatial and temporal semantic adaptations. This type of adaptation is needed in application scenarios like the one presented in Section 2.2. For instance, the adaptation in application scenario 1 is performed by combining spatial and temporal semantic reduction in order to respectively remove the unwanted objects and video segments that hold products placement.

FIGURE 3.4 – Scaling and cropping preserving ROI.

3.1.4 Adaptation Decision-taking Method

The adaptation of a video is performed in two sequential phases. The first is the adaptation decision-taking phase used to decide of the appropriate combinations of adaptation operations, find all the feasible adaptation plans and select the best plan

among the feasible ones. The second is the adaptation execution phase where the selected plan is executed over the video. Regarding the decision-taking phase, three methods have been widely investigated in the literature :

- **Knowledge-based Methods :** also referred to as multi-step adaptation methods, aim to automatically construct a suitable sequence of adaptation operations, for a given multimedia resource and a set of constraints related to the usage environments and users preferences [39]. Multimedia content adaptation frameworks implementing knowledge-based method, as in, [64], [63], [76], [13], need to have precise knowledge or information of the usage environment and user preferences, in order to reason and solve the problem of generating a suitable sequence of adaptation operations. Generally, this problem is solved using methods and techniques from the field of Artificial Intelligence (AI), such that the knowledge is represented using various representation techniques rules (e.g., procedural or declarative).
  Clearly, adaptation frameworks implementing knowledge-based methods can target Universal Multimedia Access (UMA), where any user (respectively device) can consume any multimedia content, anytime and anywhere. However, they fail to ensure Universal Multimedia Experience (UME), since they cannot provide the end-user with the best Quality of Experience (QoE). Indeed, if several feasible adaptation operations exist, knowledge-based methods are not able to select the optimal one. To this end, the Quality-based methods were introduced.

- **Quality-based Methods :** also referred to as optimisation-based methods or utility-based, aim to find the optimal selection of adaptation operations that satisfy the constraints of the usage environment and user preferences, while maximizing the quality (i.e., utility resulting from the adaptation) experienced by the end-user. The Quality-based methods were implemented by several multimedia content adaptation frameworks, as in, [127], [93], [133], [123], [104], [138], [32]. These methods operate by solving an optimisation problem. If the optimization problem involves more than one objective function to be optimized simultaneously, the problem is called multiobjective optimization or pareto optimization [144]. For instance, imagine we want to optimise both perceived quality and the execution time, which are combined in the expression of a utility function. Let us assume the existence of several feasible adaptation operations op_i and op_j, whereas the perceived quality resulting from op_i is greater than op_j, but the execution time of op_j is better than op_i. In this case, the optimum utility can be found at one of the non-dominated adaptation operations (or Pareto optimal). A solution is called Pareto optimal, if none of the objective functions can be improved in value without impairment in some of the other objective values.

- **Hybrid Methods :** combine both knowledge-based and quality-based methods in sequence. Based on the metadata description of the video, the knowledge-based method decides which adaptation operations have to be carried out in order to adapt the content according to the usage environment. Afterwards, certain intelligent adaptation tools incorporate the capability to select the parameters that optimise their output. CAIN-21 is an MPEG-21 Framework in which the adaptation engine implements a hybrid method [82].

3.1.5 Quality in Video Adaptation Context

In general, adaptation techniques cause irreversible loss of information in the generated adapted content, meaning that the original content cannot be restored from the adapted one. This information loss may reduce the quality of the delivered video, thereby minimizing the end-user's QoE. In order to ensure UME, a vital prerequisite is to design QoE-aware adaptation engine [99]. To do this, the most challenging part is to define efficient and reliable quality metrics, which are able to quantify the video quality degradation that may occur from the adaptation. Indeed, the most efficient and reliable type of quality metric is the subjective one, since it is based on scores attributed by humans who are the ultimate receivers of the adapted video [141]. Nevertheless, these methods are human resource expensive, laborious and time consuming. Therefore, the researchers have gone to design objective metrics that can predict and evaluate the quality, without human involvement and with a less cost [141]. However, it is impossible to design a quality metric that covers all the adaptation contexts, since the impact of the information loss on the end-user's QoE depends on the type of the performed adaptation. Add to that the fact that the quality experienced by the users is multidimensional in nature and can be measured on different levels [139] such as perceptual, semantic, signal, etc. (refer to Section 2.4.2). In the remainder of this section, we discuss the perceptual quality and semantic quality, which are the key aspects of the QoE in the context of video adaptation. Moreover, we highlight the difficulty of having a general quality measurement for all adaptation approaches.

3.1.5.1 Semantic Quality

'The semantic quality refers to the amount of conveyed information, regardless of how the content is presented' [126]. A high semantic quality corresponds to a high ability of the user in analysing, synthesizing and assimilating the informational content after adaptation. In order to measure the semantic quality, we need to quantify the amount of the semantic information lost during the adaptation compared to the original content. In practice, it is so difficult to come up with a general measurement of the semantic quality [104]. This is due to the following reasons :

- the difficulty in extracting semantic information from the content automatically. This is due to the well-known problem in multimedia that is, the problem of semantic gap. As defined by Smeulders et al. [120], *'the semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.'* Thus, the semantic quality is measured based on different type of semantic information, which are extracted according to the application scenario. For instance, for spatial semantic adaptation of a soccer game that involves cropping and scaling, the semantic quality can be measured by the amount of information related to the ROIs (i.e., soccer ball and players), which are maintained by the adaptation.

- the amount of semantic information conveyed by the content highly depends on the type of the content itself. For example, a news report provides different information than a soccer game.

- the loss of semantic information depends on the adaptation type. For instance,

spatial/scene summarization is performed by removing unwanted objects from the frames based on user preferences. If the removed object is semantically related to another object in the frame, then the adaptation leads to significant semantic information loss. Another example is the adaptation by spatial transcoding, which sometimes causes excessive spatial resolution reduction and impairs the details of particular frames (e.g., for a soccer game, the ball can be hardly recognized in some aerial shots.)

– the semantic quality is differently experienced from a user to another. It depends on the user's interest, cultural background, etc. For example, let us assume the case of spatial semantic adaptation of a Chinese dinner that involves frame cropping according to the actors. The quality resulting from this adaptation can be acceptable for some users who are interested in just seeing the actors. However, if the user is interested in the Chinese cultural and the way the table is set up, then the quality will be rated unacceptable.

Therefore, the measurement of the semantic quality depends on the purpose of the application scenario. In this thesis, we make use of the notion of priority, and attach it to the shots and objects (see Section 9.2.3.1). We measure the semantic quality of an adapted content with respect to the amount of semantic information, which is maintained by the adaptation. Indeed, we assume that adapting the content while preserving the highest priority items should better preserve the semantic integrity of the original video.

3.1.5.2 Perceptual Quality

'The perceptual quality refers to user's satisfaction in perceiving the content, regardless of what information the content contains' [126]. The more the distortion left by the adaptation is discernible by the human visual attention, the lower is the perceptual quality. According to the experiments conducted in [143], it appears that the human visual system perceives spatial and temporal distortions in video by performing differences in space and time. Indeed, recent neurophysiological research has found that spatial information and temporal information are processed separately in two visual pathways (ventral and dorsal streams) in the visual cortex of the human brain [43]. Whilst perceived spatial information is the amount of spatial details (e.g., shape, size, etc.) at the frame level of the video, perceived temporal information is the amount of perceived motion in the video scene. To this end, it is vital to develop perceptual quality metrics that separately measure the impact of the spatial distortion and temporal distortion on the perception of the end-user.

Similar to the semantic perceived quality, it is not possible to have general spatial and temporal perceptual quality metrics, which cover all adaptation scenarios. Depending on the type of adaptation and the video content, different parameters (e.g., frame rate, bit rate, color depth, clip type, etc.) influence the perceptual quality. A detail description of these parameters and their influence on the perception is given in [19]. For instance, consider applying the same frame rate reduction to a news report with less motion, and an action movie with a lot of motion. As the motion is known to be one of the most important visual attractors [75] [57], thus the temporal perceived quality of the adapted news report will be different than the one of the adapted action

movie. Perceptual quality has gain a lot of attention and brings to the development of several perceptual quality metrics. Recent surveys of these metrics could be found in [79] and [92].

In this thesis, we deal with a spatio-temporal semantic adaptation, which involves spatial and temporal reduction in order to remove unwanted objects and video segments. Whilst temporal reduction causes undesired temporal impairments in the video content, the spatial reduction produces undesired artifacts in the frame. Thus, we develop a temporal perceived quality and spatial perceived quality metric to measure the temporal impairments and artifacts, respectively (see Section 9.2.4).

3.2 Description for Video Adaptation using MPEG-7 and MPEG-21 Standards

Besides the design of an adaptation engine, the description of the video content and context (i.e., end-user constraints, owner constraints, terminal capabilities) is a mandatory step in video content adaptation framework. As the interoperability among the context and content descriptions is essential in adaptation frameworks to enable UMA, these descriptions should be described in some standard format. Indeed, the availability of these standardized descriptions highly contributes to the automation of the adaptation process (refer to Section 2.4.2), and alleviate the problem of extensibility and concordance with existing upcoming standards. The MPEG-7 and MPEG-21 standards address the issues associated with designing a video adaptation framework in heterogeneous usage environments, and provide a rich set of standardized descriptions and tools necessary for an interoperable adaptation framework. In this section, we overview the MPEG-7 and MPEG-21 standards with an emphasize on the tools that are related to the work of this thesis.

3.2.1 Overview of MPEG-7 Tools for Video Adaptation

MPEG-7, formally named "Multimedia Content Description Interface", is an ISO/IEC standard developed by Moving Picture Expert Group (MPEG). The goal of the MPEG-7 standard is to allow interoperable searching, indexing, filtering, and access of multimedia content by enabling interoperability among devices and applications that deal with multimedia content description [89]. Moreover, MPEG-7 aims to address as many different applications in different environments as possible, which means that it needs to provide a flexible and extensible framework for describing multimedia content [52]. To this end, MPEG-7 standardized a comprehensive set of description tools, that are :

- **D :** represent a feature, and define the syntax and semantics of the feature representation. Example of a descriptor is the Dominant Color Descriptor (DCD), which describes the representative colors distribution in an image or a region of interest through an effective, compact and intuitive representation.

- **DS :** specify the structure and semantics of the relationships between their com-

ponents, which may be both D and DS. Example of possible DSs are a movie, temporally structured as scenes and shots, including some textual descriptors at the scene level, and color and motion descriptors at the shot level.

- **Description Definition Language (DDL) :** is a language that allows the creation of new DSs and, possibly Ds. It also allows the extension and modification of existing DSs.

- **Systems Tools** : support the multiplexing of descriptions, synchronization of descriptions with the associated content, coded representations (both textual and binary format) for efficient storage and transmission, management and protection of intellectual property, etc.

These standardized tools provide support to a broad range of applications such as broadcast media selection, multimedia editing, personalized advertising, and so forth. It is important to note that the standard MPEG-7 does not specify how the descriptions are generated or how they are consumed. Only the representation format itself is specified. Further details of these tools along with some application scenarios are presented in [89] [87] [52].

The specification of the MPEG-7 standard is divided into twelve parts, among them is the Part5-Multimedia Description Schemes (MDS) [52], which provides tools that support applications requiring semantic adaptation of the video content according to user's semantic constraints, and content owner's constraints. In the remainder of this section, we overview the description tools of the Multimedia Description Schemes (MDSs) part of the MPEG-7 standard. In particular, we focus on the description tools that represent the structure and semantics of multimedia data, and the preferences of the user and owner.

3.2.1.1 Overview of MPEG-7 MDS Tools

Part5-MDS of the MPEG-7 standard specifies the MDS description tools, Ds and DSs, and provides informative examples that illustrate their use in creating descriptions. Furthermore, Part 5 provides subclauses for each MDS description tool that mainly specify their normative syntax using the Description Definition Language (DDL), and their normative semantics using text. The MDS description tools are organized on the basis of functionality as shown in Figure 3.5.

- **Basic Elements Tools :** include fundamental constructs that are used as building blocks throughout the definition of the Ds and DSs. Many of the basic elements provide specific data types and mathematical structures, which are important for audiovisual content description such as vectors, matrices and histograms. The Schema tools are intended to facilitate the formation, packaging, and annotation of MPEG-7 descriptions. The basic elements include also constructs for linking media files, localizing pieces of content, and describing time, places, persons, individuals, groups, organizations, textual annotation, classification schemes and controlled terms.

- **Content Description Tools :** describe the structure and semantics of multime-

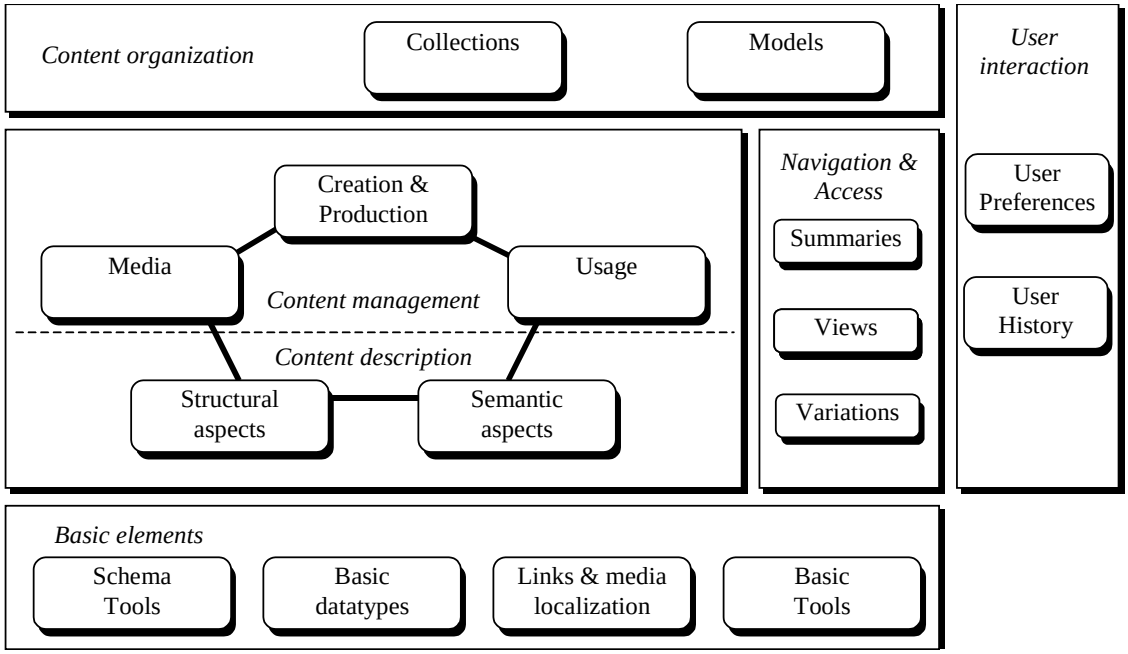


FIGURE 3.5 – Overview of the MDSs (reprinted from [52], by ISO/ IEC).

dia content. Structure description tools provide structural information on spatial, temporal or spatio-temporal components of the multimedia content in terms of video segments, frames, still and moving regions, audio segments, relations of segments, etc. Semantic description tools represent the real world related to the multimedia content, by describing semantic entities in the narrative world such as objects, agent objects, events, concepts, semantic states, semantic places, semantic times, relations of semantic entities, etc.

- **Content Management Tools :** describe information related to the creation and production processes of multimedia content (e.g., director, title, creators, creation locations and dates, short feature movie, etc.), its storage and file formats (e.g., storage location, container and encoding format, etc.), and its content usage (e.g., usage rights, usage history, broadcast schedule, etc.).

- **Content Organization Tools :** organize and model collections of multimedia content. The collection tools are used for tasks such as describing an album of photos, or a cluster of color feature descriptors. The model tools describe parameterized models of multimedia content, descriptors, or collections. The models can be expressed in terms of statistics or probabilities associated with the attributes of collections of multimedia content.

- **Navigation and Access Tools :** facilitate browsing and retrieval of audiovisual content by defining summaries, views and partitions, and variations of multimedia content. The summary tools describe semantically meaningful summaries of multimedia content to enable efficient browsing and navigation. Moreover, the view tools describe structural views of the multimedia signals in the space and/or frequency domain to enable multi-resolution access and progressive retrieval. Finally, the variation tools describe relations between different variations of multimedia content to allow adaptive selection under different terminal, delivery, and user preference conditions (see example in Section 3.1.3.1).

– **User Interaction Tools :** describe the user preferences pertaining to the consumption of multimedia content, as well as the user history in consuming the content (i.e., usage history). The availability of these descriptions allows a personalized selection and consumption of multimedia content, by matching the MPEG-7 multimedia content descriptions to the user preferences descriptions.

The most important MPEG-7 MDSs for the purpose of video semantic adaptation applications are included in the content description and user interaction groups, which will be presented with more detailed in the remainder of this section. In addition, we overview the classification schemes and controlled terms tools, which enable the representation of semantics and controlled vocabularies.

3.2.1.2 Content Description Tools

The content description tools are used to describe perceivable information of the multimedia content, comprising structural aspects (structure description tools) and conceptual aspects (semantic description tools).

Structure description tools ; they are used to describe the structure of a multimedia content in time and space. They are based on the concept of a "segment" defined by the Segment DS that generally refers to a spatial and/or temporal unit of multimedia content (e.g., temporal segment of video may correspond to a set of shots or a group of scenes). The structure description tools are used to describe the result of a spatial, temporal, or spatio-temporal partitioning of a multimedia content (e.g., spatial regions of an image, temporal segments of a video, etc.), as well as a full multimedia content (e.g., entire video stream). They can also be used to describe segment properties (e.g., location, start time and duration, etc.), structural decomposition, and structural relations among segments. The structure description tools are organized on the basis of their functionalities as follows :

– **Segment entity description tools :** provide specialized segment description tools extended from Segment DS, in order to describe the structure of specific types of multimedia segments. These tools include, but not limited to the following : VideoSegment DS, for describing temporal intervals or segments of the video ; StillRegion DS, for describing a 2D spatial regions of an image or video frame ; MovingRegion DS, for describing a 2D moving region of a video segment ; and Mosaic DS, for describing a mosaic or panoramic view of a video segment.

– **Segment attribute description tools :** describe attributes of segments related to creation and media information (i.e., CreationInformation DS and MediaInformation DS), spatio-temporal decomposition (i.e., SpatialMask D, TemporalMask D, etc.), the importance of segments from a specific viewpoint (i.e., PointOfView D), specific media and creation information of an ink segment (i.e., InkMediaInfo DS), information about a hand writing recognizer (i.e., HandWritingRecogInfo DS), etc.

– **Segment decomposition tools :** describe structural decomposition of segments of multimedia content. MPEG-7 has defined four types of segment decompositions : spatial, temporal, spatio-temporal, and media source decompositions. For

instance, an image can be spatially decomposed into a set of still regions corresponding to objects within the image (i.e., spatial decomposition), and the still regions can be in turn decomposed into other still regions (i.e., media source decomposition).

- **Segment relation description tools :** describe structural relations among segments of multimedia content. MPEG-7 has defined four types of segment relations : spatial, temporal, spatio-temporal, and others. Current normative segment relations in MPEG-7 are Allen's temporal interval relations (e.g., before, after, etc.) [6], directional spatial relations (e.g., left, above, south, north, etc.), topological spatial relations (e.g., touch, overlap, etc.), n-ary spatio-temporal relations (e.g., union, intersection, etc.), and other binary relations (e.g., keyFor, annotatedBy, etc.).

Semantic description tools ; they are used to describe the narrative world, which refers to the reality in which the description makes sense, such as background, context and participants that are depicted in, or related to multimedia content [52]. These tools represent the narrative world by describing semantic entities such as objects, events, concepts, states, places, and times, together with their attributes and relations in those narrative worlds. The semantic description tools are grouped into the following categories :

- **Semantic entity description tools :** provide specialized semantic entity description tools to describe specific types of semantic entities, which exist or take place in narrative worlds. These tools include, but not limited to the following : Object DS, for describing perceivable objects that exist and take place in time and space in the narrative world (e.g., Mozart's piano), or abstractions of perceivable objects (e.g., any piano) ; AgentObject DS, for describing an individual person, a group of persons, or an organization ; Event DS, for describing perceivable events that take place in regions in time and space of a narrative world (e.g., Mozart is playing piano), or abstractions of perceivable events (e.g., anyone is playing piano) ; and SemanticPlace DS and SemanticTime DS, for describing locations and times in the narrative world, respectively.

- **Semantic attribute description tools :** describe other attributes of semantic entities related to abstraction levels and semantic measurements in time and space. Abstraction refers to the process of taking a concrete semantic description of specific multimedia data (e.g., the description "Mozart is playing piano" for a specific image), and generalizing it to be applicable to any multimedia data (media abstraction, e.g., the description "Mozart is playing piano" for any image with no links to a specific image), generalizing it to a set of concrete semantic descriptions (formal abstraction, e.g., "Someone is playing piano" or "A man is playing piano").

- **Semantic relation description tools :** describe semantic relations such as the relationship between two objects (MemberOf relation, e.g., an object "player" is a member of an object "team"). MPEG-7 has defined normative semantic relations including, but not limited to the following : relations among semantic entities (similarTo relation, e.g., the object "man" is similar to object "human") ; relations among segments and semantic entities (mediaPerception Of relation, e.g., the image "Mozart.jpg" that shows "Mozart" is a media perception of objects

"Mozart"); relations among analytic models and semantic entities (symbolOf relation, a picture of "zeros and ones" can be a symbol of object "computers").

For MPEG-7 structural and semantic description examples, we refer the reader to Annex B.

3.2.1.3 User Interaction Tools

In a multimedia system, user modeling is also needed along with the content modeling to enable effective user interaction, and personalized access and consumption of multimedia content. As defined in [39], user modeling refers to building a profile of the user's preferences for consumption and usage. For this purpose, MPEG-7 MDS specifies two tools related to user interaction with multimedia content, which are the UserPreferences DS and the UsageHistory DS.

UserPreferences DS ; it is used to describe user's preferences pertaining to consumption of multimedia content. To begin with, UserPreferences DS contains basic tools to enable users to specify their preferences for certain types of content, and for ways of browsing the content. The UserIdentifier datatype is used to identify a particular user preference description and distinguish it from other user preferences descriptions. The PreferenceCondition D allows the users to condition the use of their preferences in a particular context, in terms of time and place (e.g., a user may have preference for news in English language when he/she is travelling in Germany). The userChoice datatype is used to give the users explicit control of the preference description (e.g., a user may indicate that his/her preferences should not be communicated to a service provider). The preferenceValue datatype is used to allow users to specify the relative importance of their preferences on a scale of -100 to 100 with respect to other preferences, in case multiple preferences of the same type are present. For instance, a user may like both "science-fiction" and "romantic" genres of movies, but may prefer the latter over the former. In this case, the user specifies a higher preference value to "romantic" genre movies than the "science-fiction" one. Furthermore, UserPreferences DS contains tools to specify user's preferences pertaining to filtering, searching and browsing of multimedia content :

- **FilteringAndSearchPreferences DS :** describes user preferences for filtering or searching for preferred multimedia content. These preferences are only concerned with the creation, classification and source aspects of the content :
  - *CreationPreferences DS* describes preferences related to the creation description of the content such as preferred title or creator of the content, preferred period of time or location where the content was created and user's preferred content through the use of textual keywords.
  - *ClassificationPreferences DS* describes preferences related to the classification of the content such as preferred country of origin and period of time when the content was first released, preferred genre (e.g., sports, politics, etc.) and language of the content, etc.
  - *SourcePreferences DS* describes preferences related to the media source such as a preferred delivery mechanism of content (e.g., web-cast, streaming, etc.), preferred place and date where and when it is made available for consumption, preferred distributor or publisher, preferred format for the media, etc.

– **BrowsingPreferences DS :** describes user preferences pertaining to navigation of and access to content. These preferences may be conditioned on certain times and locations, and type of multimedia content in terms of genre. For instance, a user may wish to browse only the multimedia content matching the "politics" genre, which where released in 2010. Furthermore, BrowsingPreferences DS specifies preferences through the SummaryPreferences DS that are related to media summaries and their visualization. Indeed, the SummaryPreferences DS describes preferences, which concern the summary type and theme, minimum/maximum/ preferred AV summary duration, minimum/ maximum/preferred number of key frames in a visual summary, minimum/maximum/preferred length, in number of characters, of a textual summary, etc.

UsageHistory DS ; it is the MPEG-7 logging tool for user interactions. It serves as a container DS for UserActionHistory elements and a UserIdentifier element, which are used to describe the history of actions carried out by a specific user over a multimedia content during an observation period. The UserIdentifier identifies the user for whom the usage history is provided. The UserActionHistory describes the multiple lists of actions performed by the user over one or more, non-overlapping observation periods. Each action list contains a specific type of user action (e.g., record, pause, play, etc.), regarding multimedia content. Moreover, every user action is associated with the time and duration of the action, an identifier of the multimedia content for which the action took place, and optional referencing elements that can point to a part of the content description or other related material. Regarding the collection and the exchange of usage history information, the UsageHistory DS allows the user through the attribute allowCollection to determine whether his/her consumption history can be tracked, collected or distributed. Furthermore, this DS enables the user to specify whether his/her identity should be kept private or can be revealed to third parties.

The collection and representation of usage history information in a standardized format, as well as the users preferences, are relevant to various application areas. For instance, the existence of users preferences descriptions is a vital prerequisite for multimedia recommendation systems. Nevertheless, it is difficult to ask the users to explicitly give their preferences since it requires effort from them that they are not willing to make. This is particularly true when regular updates of users preferences are needed as their preferences may change over the time. To this end, some multimedia recommendation systems collect and analyze usage history information, in order to infer users preferences with regard to multimedia content. Further application scenarios benefiting from the usage of the user interaction tools are found in [87].

3.2.1.4 Classification Schemes and Terms

MPEG-7 MDS specifies tools for representing Classification Schemes (CS)s, defining terms inside CS, and for using terms inside descriptions. A CS is a set of standard terms that form a vocabulary for a particular application or domain. A term represents a well-defined concept in the domain covered by the CS. A term has an identifier that uniquely identifies it within a CS, a name that may be displayed to a user or used as a search term in a target database, and a definition that describes its meaning. Moreover, a CS may organize the terms that it contains with a set of term relations. As defined in the

standard [52], "a term relation is a relation between two terms in a CS, such as whether one term includes the meaning of another term. In most cases, the primary relation between terms will be to indicate whether one term is 'narrower than' or 'broader than' another term in meaning. When terms are organized this way, they form a classification hierarchy". For more information about the possible relationships between two terms, we refer the reader to the subclause defining the ClassificationScheme DS in [52]. An exemplary instance for the products placement classification scheme is depicted in Listing 3.1, where each term is defined by an identifier, a name and a definition.

Listing 3.1 – Products Placement CS.

```
...
   <mpeg7:Description xsi:type="mpeg7:ClassificationSchemeDescriptionType
       ">
      <mpeg7:ClassificationScheme uri="urn:ProductPlacementCS">
         <mpeg7:Term termId="1">
            <mpeg7:Name xml:lang="en">Computer</mpeg7:Name>
            <mpeg7:Definition xml:lang="en">Product Placement for a
                computer</mpeg7:Definition>
            <mpeg7:Term termId="1.1">
               <mpeg7:Name xml:lang="de">HP_Computer</mpeg7:Name>
               <mpeg7:Definition xml:lang="en">Product Placement for an
                   HP_Computer</mpeg7:Definition>
            </mpeg7:Term>
            <mpeg7:Term termId="1.2">
               <mpeg7:Name xml:lang="en">Apple_Computer</mpeg7:Name>
               <mpeg7:Definition xml:lang="en">Product Placement for an
                   Apple_Computer</mpeg7:Definition>
            </mpeg7:Term>
            ....
```

Regarding its use inside a description, a term in a CS is referenced with the TermUse or ControlledTermUse datatype. Whilst the former allows a term to be either referenced or written in free text, the latter controls the use of terms from a CS by requiring a reference to a term. This reference is included in the href attribute, which is defined as an optional (resp. required) attribute in the TermUse datatype (resp. ControlledTermUse datatype). If href is absent, then the term is defined directly within the TermUse instance using free text. Otherwise, different possibilities exist for referring to terms. Indeed, href is of type termReference datatype, which supports two forms for term referencing : 1) a Uniform Resource Identifier (URI) reference and 2) aliased term reference that is an abbreviated form for referencing terms in CSs that have been assigned aliases (see Listing 3.2).

Listing 3.2 – Examples of term reference using TermUse (reprinted from [52], by ISO/IEC).

```
<!-- Referring to a term using a URN -->
<Genre href="urn:mpeg:GenreCS:alternativeJazz"/>

<!-- Referring to a term using a HTTP URI -->
<Genre href="http://www.mpeg.org/GenreCS.xml#alternativeJazz"/>

<!-- Referring to a term using a classification scheme alias -->
<!-- Define schema aliases -->
<ClassifiationSchemeAlias alias="s1" href="urn:mpeg:GenreCS"/>
<ClassifiationSchemeAlias alias="s2" href="http://www.mpeg.org/GenreCS.
    xml"/>

<!-- Refer to the term using the two schema aliases -->
```

```
13 <Genre href=":s1:alternativeJazz"/>
14 <Genre href=":s2:alternativeJazz"/>
```

The first example depicts the use of a Uniform Resource Name (URN) of the form "urn: mpeg: scheme: termId", as the term reference URI. As defined by MPEG-7, the URI becomes a URN whenever the term being referenced is defined in ISO/IEC 15938. In this URN, the "urn: mpeg: scheme" part that identifies the CS, should be exactly equal to the value of the URN value specified in a ISO/IEC 15938 CS. More, the "termID" part that identifies the term, should match one of the "termId" defined in the identified CS. Even if the CS are not defined in ISO/IEC 15938, using a URN as the term reference URI is strongly recommended by the standard [52]. Besides URN, a URI can use an HTTP format for referring to a term, as shown in the second example. The last two examples show the use of the CSs aliases. For instance, the alias "s1" is assigned to the CS identified by "urn: mpeg: GenreCS" (line 9), and it is used in line 13 to refer to the term "alternativeJazz" in this CS. For further details on term references see the specification of the termReference datatype in [52].

3.2.2 Overview of MPEG-21 Digital Item Adaptation

MPEG-21 is an emerging ISO/IEC standard that aims at defining a multimedia framework to enable transparent and augmented use of multimedia content across a wide range of networks, devices and user preferences. The framework is intended to cover the entire multimedia content delivery chain encompassing content production, protection, adaptation and delivery. Digital Item (DI) is the fundamental unit of distribution and transaction within this framework. This entity is defined as structured digital objects, consisting of the media content itself and its corresponding metadata. A major part of these standardization efforts is Part 7 of MPEG-21, referred to as DIA [55]. Figure 3.6 illustrates the concept of DIA : a DI is subject to both resource adaptation and descriptor adaptation engine, which together produce the adapted DI. As shown in the figure, the standard specifies only the tools that are used to guide the adaptation engine, whereas the adaptation engines themselves are left open to various implementations. The DIA tools are clustered into eight major categories [55]. Nevertheless, the purpose of this section is not to cover in-depth all the various tools and their specifications. Our emphasize will be rather in reviewing the Universal Environment Description (UED) tools in particular the User Characteristics Tools, and the Universal Constraint Description (UCD) tools that are relevant for video semantic adaptation applications. These tools are explained in the remainder of this section. For further information about them, we refer the reader to the Part 7-DIA [16].

3.2.2.1 Universal Environment Description (UED) Tools

The MPEG-21 UED tools are used to describe user characteristics, terminal capabilities, network characteristics as well as the natural environment in which the DI is finally to be consumed (see Figure 3.7).

The UED tools are organized on the basis of their functionalities as follows :

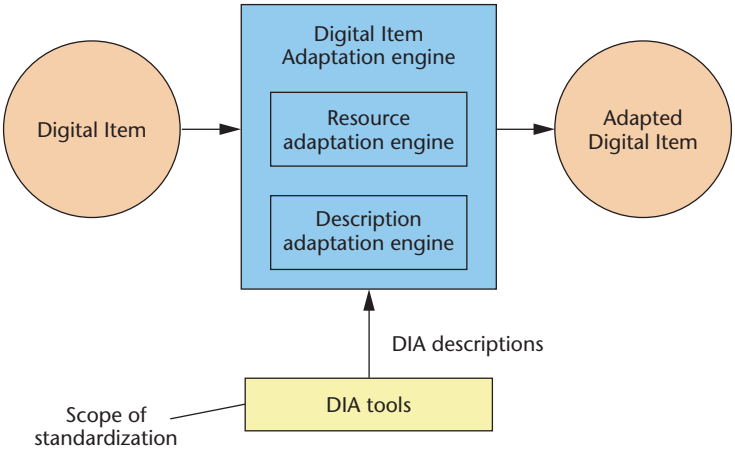


FIGURE 3.6 – Concept of MPEG-21 DIA (reprinted from [55], by ISO/ IEC).

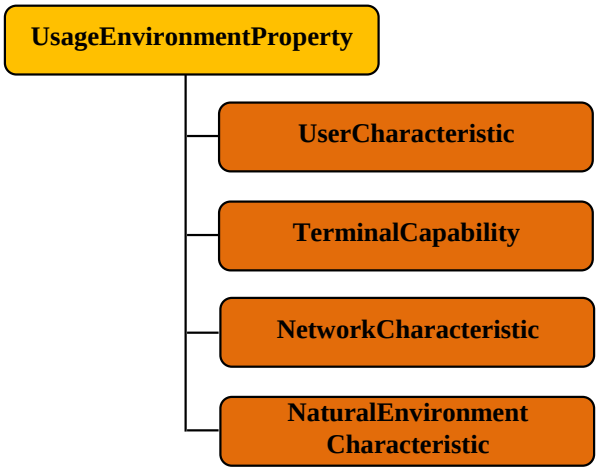


FIGURE 3.7 – UED tools (reprinted from [67], by E. Kasutani).

- **User Characteristics Tools :** enable the description of various characteristics of the user, including general user information, usage preferences and usage history, presentation preferences, accessibility characteristics, and location characteristics. These descriptions are relevant for various applications concerned with semantic- and technical-driven adaptations such as adaptive selection, personalization of content, and so forth. The user characteristics tools include several tools imported from MPEG-7 MDS, as well as a number of newly developed ones.
  - *UserInfo Tool :* describes general characteristics of the user, such as name and contact information, where a user can be a person, a group of persons, or an organization. DIA imports the Agent DS from MPEG-7, in order to specify the general information about a user.
  - *UsagePreferences and UsageHistory Tools :* the former tool describes the usage preferences of a user related to the type and content of DIs, and is derived from UserPreferences DS specified in MPEG-7 (refer to Section 3.2.1.3). The latter tool describes the history of the actions done by the user on DIs, and is derived from UsageHistory DS specified in MPEG-7 (refer to Section 3.2.1.3).
  - *Presentation Preferences Tools :* define a set of preferences related to the means by which DIs and their associated resources are presented or rendered to the user. This set includes, but not limited to the following : AudioPresentation-Preferences regarding the presentation or rendering of audio resources (e.g., preferred volume, preferred frequency equalizer settings, etc.) ; DisplayPresen-

tationPreferences regarding the presentation or rendering of images and videos (e.g., preferred color, preferred brightness or saturation, etc.) ; GraphicsPresentationPreferences related to graphics media (e.g., preferred degradation of geometry for graphics, preferred animation, etc.) ; ConversionPreference regarding the types of conversions that a user prefer (e.g., a user can apply some conversion rules to video resources, such as in case modality conversion is needed because the terminal can only support the transport of image and text modality, the video should be converted to audio as a first preference, or to image or text as a second preference.).

  - *Accessibility Characteristics Tools :* include AuditoryImpairment and VisualImpairment tools that provide descriptions, which enable one to adapt content according to certain auditory or visual impairments of a user. The AuditoryImpairment tool describes the characteristics of a particular user's auditory deficiency (e.g., hearing thresholds of a user at various frequencies in the left and right ear). The VisualImpairment tool describes the characteristics of a particular user's vision deficiency (e.g., total blindness, color vision deficiency, i.e., the inability to recognize certain colors, etc.). These descriptions could be used in applications requiring perceptual-driven adaptation.
  - *Location Characteristics Tools :* include MobilityCharacteristics and Destination tools that describe the mobility and destination of a user, respectively. The MobilityCharacteristics tool provides a concise description of the movement of a user over time, particularly information about directivity, location update intervals and degree of randomness in user movements. The Destination tool describes the destination of a user (e.g., name and geographical location of the user destination). With these descriptions, adaptive location-aware services could be provided.

- **Terminal Capabilities Tools :** describe both receiving and transmitting capabilities of a terminal in terms of encoding and decoding capabilities, display and audio output capabilities, power, storage, and input/output characteristics of a device. These tools provide the required information in order to perform a technical-driven adaptations. For example, information regarding the format and the bit-rate at which the content could be playable on the user's terminal.

- **Network Characteristics Tools :** include description of static capabilities of a network such as its maximum capacity, as well as dynamic conditions of a network such as the available bandwidth, error, and delay. These tools provide the required information in order to perform technical-driven adaptations.

- **Natural Environment Tools :** pertain to physical environmental conditions such as the lighting conditions, auditory noise level, or a circumstance such as the time and location that content is consumed or processed. These tools provide the required information in order to perform perceptual-driven adaptations.

3.2.2.2 Universal Constraint Description (UCD) Tools

The standard MPEG-21 specifies the UCD tools for describing constraints for adaptation. The UCD allows the consumer of the multimedia content, i.e., end-user, to provide constraints to the adaptation engine. In this case, the UCD supplements the

information in the usage environment descriptors and also converts it into a semantics free form for format-independent decision-making. For instance, the end-user may define a constraint where the images after an adaptation should match the color capability of the terminal. Furthermore, the UCD allows the provider to constrain the usage and usage environment of a Digital Item by means of limitation and optimization constraints. For instance, the usage of a DI containing an image resource can be constrained by the content provider such that, the resolution of the image after an adaptation should not be smaller than 40% of the receiving terminal display resolution. The content provider can also formulate some optimization constraints for the adaptation, such as the image should be at least smaller than 50% of the rendering terminal display resolution, because the rendering application should not run in full-screen mode to conserve resources. More details about the UCD tools can be found in [55].

3.3 Conclusion

In this chapter, we presented the most important concepts related to video content adaptation and discussed the requirements to design an adaptation engine. We have discussed the different adaptation approaches that have been proposed in the literature and present their classification. Moreover, we have discussed the existing adaptation decision-taking methods and argued that the quality-based adaptation decision-taking methods ensure the UME vision. Accordingly, we have discussed both semantic quality and perceptual quality, and highlighted the difficulty of having a general quality measurement for all adaptation approaches. Finally, an overview was given on some parts of MPEG-7 and MPEG-21 on which the presented work has been directly founded, and to which it has contributed. Indeed, a good understanding of MPEG-21 DIA and MPEG-7 MDS is required for the comprehension of this dissertation.

Chapter 4

Analysis of existing Video Adaptation Frameworks

The objective of this chapter is to review the existing video content adaptation frameworks, and analyze their possible adoption for the purpose of this thesis. To begin with, we list the criteria of the analysis using the concepts defined in Chapter 3. Then, we describe each framework and discuss the design of its adaptation engine. Furthermore, we study the feasibility and limitations of these frameworks in performing video adaptation according to the requirements of the thesis. We conclude this chapter with a positioning of the frameworks and a discussion assessing the need of developing the Personalized vIdeo Adaptation Framework (PIAF).

4.1 Analysis Criteria for Video Adaptation Frameworks

Adaptation frameworks proliferate in the form of retrieval, summarization, adaptation or personalization systems. The design of their adaptation engine is strongly dependent on the requirements of the UMA application in question (refer to Section 3.1). Thus, a straightforward comparison of the adaptation frameworks is senseless, except if they target the same application scenario. To this end, we opted instead to an analysis of mostly known video adaptation engines according to the requirements of the application scenarios discussed in Section 2.3 and 2.4 of Chapter 2. This analysis is performed with respect to the following criteria :

- Granularity level : What is the granularity level that the adaptation engine deal with ? Does it deal with a fine granularity down to the object level ?

- Adaptation operation : Which type of adaptation operation can the adaptation engine perform ? Is the removal operation included ?

- Adaptation approach : Is the adaptation approach semantic-driven, perceptual-driven or technical-driven ?

- Adaptation decision-taking method : Is the adaptation decision-taking method knowledge-based, quality-based or hybrid ?

- QoE-awareness : Is the adaptation engine aware of the user's QoE ? Does it consider the semantic quality (SemQ) and the perceptual quality (PQ) ?

- QoE-awareness : Is the adaptation engine aware of the key aspects of the user's QoE, which are the semantic quality (SemQ) and the perceptual quality (PQ) ?

- Adaptation focus : Who is the target of the adaptation, the user or the terminal ?

- Conflicting constraints : Does the adaptation process deal with conflicting constraints ?

- Owner intellectual property rights : Does the adaptation framework enable the owner to formulate constraints on how their videos could be manipulated ?

4.2 Analysis of Individual Frameworks

Among the adaptation frameworks reported in the literature, we overview the following ones : koMMa [63] [64], CAIN [82], DCAF [12] [123], NinSuna [133], DANAE [106] and SAF [149]. In the remainder of this section, we describe these frameworks individually and consider the pros and cons of each one.

4.2.1 koMMa : Knowledge-based MultiMedia adaptation

The koMMa framework [64], [63], [76] has been designed to perform content adaptation by composing various multimedia adaptation tools. The motivation behind the conception of koMMa is to address the issues of openness, extensibility, and concordance with existing and upcoming standards. Indeed, the authors argue that it is does not seem realistic for one single software tool to perform all required adaptation steps for the various user preferences, terminal capabilities, network characteristics, or even for the diverse set of coding formats. Therefore, there is a need of having an open and extensible adaptation engine, such that no changes in the general mechanism are required when new forms of adaptation are possible as the standards evolve or new tools become available.

Based on this argument, the authors conceived the koMMa framework that is capable of computing and executing multi-step adaptation sequences based on semantic descriptions of the available adaptation operations. The problem of the adaptation decision phase that is, constructing adequate adaptation sequences, is viewed as a state-space planning problem [111]. State-space planning is a classical technique from the field of Artificial Intelligence (AI) that enables an intelligent agent to accomplish automatic decisions before acting. Such a planning problem consists typically of passive entities (i.e., start state and goal state) and active entities (i.e., actions). The start state corresponds to the description of the current format of the multimedia content, which is represented by the MPEG-7 Variation Tools. The goal state corresponds to the description of the format to which the multimedia content should be adapted, in order to fit the user's UED (terminal's capabilities, supported codec, spatial resolution, etc.). The actions are the adaptation operations applied on an initial state to reach a goal state. Their semantics is captured in the form of input, output, precondition, and

effects (IOPE) (see examples in [64]), and use the standardize OWL-S for declaratively representing the IOPE of a service. Moreover, the adaptation decision in koMMa is done automatically since it considers deterministic adaptation operations, which implies that the output state can precisely be determined from the input state [81].

The major novelty of the koMMa framework is the use of a knowledge-based method, which computes adaptation plans independent of specific tool implementations [13]. Therefore, no changes in the general adaptation mechanism are required when new type of adaptation operations are added. Furthermore, koMMa ensures the interoperability between its modules by making use of the MPEG-7 and MPEG-21 description tools.

However, several issues still need to be addressed in the koMMa framework. First, since the adaptation engine is dependent of the UED descriptions, then the user can specify the best goal state - maximum display resolution, the highest frame rate and bit rate - which a terminal is not able to deal with. In many situations, such an adaptation is not possible and koMMa fails to construct a plan. Moreover, koMMa is based on the assumption that the behavior of the operations is well known before taking decisions. Nevertheless, koMMa does not consider the execution order of the operations in case of several ones. For instance, in case of a plan consisting of spatial reduction and grayscale color transformation, it could be better to first reduce the image size and then apply the grayscale color transformation. In koMMa, the optimization strategies are not considered. Furthermore, since the planning algorithm is deterministic, koMMa can deal only with one initial state described without ambiguity in the UED. Thus, in case of several initial states, no plan is constructed. Also, koMMa cannot efficiently deal with several goal states. If this latter situation occurs, a goal state is selected arbitrarily with no guarantee to be the best choice. As a concluding fact, koMMa is a non-utility aware multimedia framework, and the adaptation is not aware of the end-user's QoE.

4.2.2 CAIN : Content Adaptation INtegrator

'CAIN-21 is a multimedia adaptation engine that facilitates the integration of pluggable multimedia adaptation tools, automatically chooses the chain of adaptations to perform and manage its execution' [82]. Similar to koMMa, CAIN provides a set of well described adaptation tools, so called Content Adaptation Tools (CATs). CATs are pluggable software adaptation into an MPEG-21 compliant adaptation engine. They allow the integration of semantic- and technical-driven adaptation approaches : transcoding, transmoding, scalable content and temporal summarization. Among the CATs integrated in CAIN [130], there is the semantic-driven adaptation tool *image2videoCAt* that involves a transformation of an image to a video presentation while considering user's ROI [83], [90].

Moreover, the adaptation engine of CAIN implements a knowledge-based method. Yet, CAIN remedies to the problem of multiple initial states mentioned above in the description of koMMa, and accepts two UEDs from the user [84] [81] : mandatory UED and desirable UED. The former contains the constraints related to the terminal capabilities and resource limitations that must be definitely fulfilled. The latter contains the users preferences that should be satisfied. The adaptation decision-taking algorithm of CAIN begins by satisfying the constraints of the mandatory UED. If more than one

adaptation tool satisfy these constraints, then the desirable constraints are applied to reduce the solution space. If the solution space is not yet reduced to one adaptation solution, then a final optimization step chooses the feasible solution. Indeed, this solution is obtained by using the VQM [100] and PSNR [9] quality metrics to create a ranking between the adaptation tool.

The novelty of CAIN is in combining knowledge- and quality-based decision-making methods in order to find a solution when there is one. However, CAIN is based on the assumption that the user knows his/her preferences for a given content in advance and can directly provide the desirable UED when it is needed. Nevertheless, this assumption is not true in the real world where the user cannot know the best spatial resolution, frame rate, etc. in a specific consumption scenario where different devices, network characteristics, etc. exist. Therefore, the users are in general not able to provide a reliable UED that would represent their preferences or requirements precisely enough. As a consequence, in case of more than one adaptation strategies, CAIN will not be able to choose the adaptation plan that suits best the usage context constraints.

4.2.3 DCAF : Distributed Content Adaptation Framework

DCAF is a software architecture of multimedia content adaptation that provides a distributed content adaptation approach for infrastructure-based pervasive computing environments [12] [123]. The objective of DCAF is to provide a general content adaptation solution offering flexibility, scalability, extensibility and interoperability. Indeed, unlike the existing approaches that have been proposed for content adaptation namely sever-based, client-based and proxy-based, the approach used by DCAF architecture is service-based. The proposed adaptation approach uses Internet accessible adaptation services to carry out the content transformation. In this approach, content adaptation is performed by composing adaptation services available on the Internet(i.e., Web services).

Among the components of the DCAF architecture, the main component is the Local Proxies (LPs) on the client-side. They are in charge of retrieving and processing the context profile when a user makes a request. Afterwards, the LPs find, compose and execute the adaptation services in order to perform content adaptation. The core part of the local proxy is the Content Negotiation and Adaptation Module (CNAM), which has two main components : the adaptation decision engine and the adaptation planner. For the remainder of this section, we call on readers to draw attention to the ambiguous use of the latter terminologies in the context of DCAF.

The adaptation decision engine identifies the types of adaptation to perform, by using the environments parameters : user context (e.g., user's preferences, location), device context (e.g., screen size), network context (e.g., bandwidth), and content metadata. The output of the adaptation decision process is a transformation list (i.e., list of adaptation tasks), which serves as an input for the adaptation planner. The latter is concerned with the selection and the composition of the appropriate adaptation services. Since more than one service can fulfill the same adaptation tasks, the selection is based on QoS criteria to determine which services would be more suitable to fulfill the required adaptation tasks. The QoS model used in [12] contains two criteria : the response time of the service that consists of the execution and transmission time, and

the cost of the service that consists of the service execution and transmission charge. In order to generate an adaptation execution plan, the adaptation planner constructs a graph called adaptation graph. It represents all service compositions that the content adaptation process can perform. The nodes of the graph represent adaptation services, while the edges represent the possible connections between adaptation services. Once the graph is constructed, the optimal service composition plan (also called the optimal composite service) is found by applying Dijkstra's algorithm [28]. Indeed, the choice of the best alternative in the graph (also called the optimal path), is done based on user specified QoS criteria.

The major novelty of DCAF is the use of the adaptation services for content adaptation, which makes from DCAF a flexible, scalable, extensible and interoperable framework. Indeed, the use of a Service-Oriented Architecture (SOA) enables the usability of the adaptation tools by different application scenarios. Moreover, it is efficient that the services are developed to implement a particular adaptation independently of the application scenario. Thus, it is possible to integrate a new adaptation tool, adapt the code of an existing one, or even use other tools with the same functionalities that provides better performance. Furthermore, DCAF implements a quality-based decision-making method to deal with the problem of having several services for the same adaptation tasks. The choice of the best adaptation service is obtained by using the QoS criteria quality , i.e., the service response time and the service price.

Compared to the CAIN framework, it can be drawn that the application scenarios covered by DCAF could also be covered by CAIN [82]. Similar to the adaptation services, CATs are also independent from the application scenario since they are pluggable software adaptation. Furthermore, the adaptation engine of CAIN is capable of finding the optimal service composition plan by formulating the QoS criteria as users preferences that should be satisfied, and integrating them in the desirable UED. Nevertheless, the same drawbacks related to the completeness of the adaptation solution in CAIN, is applied for DCAF. Indeed, the adaptation is static as the composition of available services is generated automatically by the adaptation planner only if the user specify the inputs and outputs required by the composite server.

4.2.4 DANAE : Dynamic and distributed Adaptation of scalable multimedia coNtent in a context Aware Environment

DANAE is an IST[8] European co-funded project [24]. DANAE implements and extends the existing MPEG-21 adaptation mechanisms to ensure UMA. Its objectives are to specify, develop, integrate and validate in a realistic testbed a complete framework (with servers, network devices and terminals) for context-aware, dynamic and flexible media adaptation, delivery and consumption, with the ability to provide end-to-end quality of multimedia service at a minimal cost to the end-user [106]. To do this, the project proposes a platform for dynamic and distributed adaptation of scalable multimedia content in a context-aware environment. DANAE uses a distributed content adaptation approach. In this approach, the adaptation process is performed in a distributed fashion along the path between the client and the server, thus reducing the

8. IST : Information Society Technology

computational load on the server.

As outlined by the authors in [106], the DANAE project resulted in three different MPEG-21 based adaptation approaches that were later standardized : (1) Digital Item Processing (DIP) enables static stream selection at the start of a session and in session mobility scenarios [54] ; (2) Resource Conversion in MPEG-21 DIA enables dynamic stream selection at any time in a stream [69]. Alternatively, another tool for multimedia resource adaptation within MPEG-21 was developed that is the Bitstream Syntax Description (BSD) tool [40]. BSD-based adaptation enables fine-grained and dynamic scalability of multimedia content in a generic way as the high-level structure of the bitstream is described with a BSD document in XML ; (3) finally, distributed adaptation extends the BSD-based adaptation approach in order to enable this adaptation mechanism anywhere along the delivery chain. All of these adaptation approaches were researched and implemented, particularly an interactive and user-centric framework called Semantic Adaptation Framework (SAF) [149] that deals with video personalization. In the remainder of this section, we describe the SAF framework and consider its pro and cons.

4.2.4.1 SAF : Semantic Adaptation Framework

As discussed in Section 2.4.2, video personalization requires the existing of semantic user preferences and semantic description of the video content. Moreover, the availability of standardized descriptions allows intelligent interaction over the content such as automatic content filtering according to semantic user preferences. SAF addresses these requirements by generating all required semantic content annotation and context representation in MPEG-7. It also enables an MPEG-21 adaptation engine to semantically adapt the video content, and therefore enhance the user experience. Indeed, SAF integrates MPEG-7 semantics description tools into the MPEG-21 Multimedia Framework to enable summarization of spatial and temporal properties of scalable media. SAF consists of the three following modules :

1. **SemanticGenerator :** it generates MPEG-7 compliant ontologies for content annotation and context representations. It also consists of a Semantic Annotation Tool (SAT) that provides the semantic annotation of MPEG-4 videos in terms of semantic entities (e.g., events, concepts, objects, people, location, time) and their relations based on the existing ontologies. Moreover, SAT generates semantic generic Bitstream Syntax Descriptions (gBSDs) of the video, which consist of gBSDs [55] indexed with semantic metadata.
2. **Semantic User Preferences Tool :** it allows a user to select her/his preferred topics, content structure and semantic entities with their relations from the available ontologies.
3. **Semantic Adaptation Engine :** it consists of two components : the description adaptation engine and the resource adaptation engine. Upon a user request for a video content, the description adaptation engine computes an adaptation decision based on the user preferences and the semantic Adaptation Quality of Services (AQoSs), which define all the possible semantic adaptations for a given video. The adaptation decision consists of a parameterized XSLT style sheet, which may be also created by the user itself. Afterwards, this XSLT style sheet is used to transform the semantic gBSD in order to fit the user preferences. Once it is transformed, the semantic gBSD is sent to the resource adaptation engine

to adapt the video in question.

The major novelty of SAF lies in the combined use of MPEG-7 and MPEG-21 DIA descriptions. This combination enables adaptation by temporal and spatial/scene summarization. Indeed, the MPEG-7 semantic description tools provide means for a semantic description that is close to the human understanding of multimedia content. Thereby, by integrating these tools into the MPEG-21 multimedia framework, this enables SAF to make semantic adaptation of the video, and thus maximizing the user experience. Besides being a user-centric framework, SAF architecture is modular and easily allows extensibility and scalability of both software modules and semantic metadata. Moreover, the computation of the adaptation decision in SAF used neither knowledge-based nor quality-based methods. The summarization is simply done by filtering the content over the context. For instance, given an MPEG-4 video content, the semantic adaptation engine transform its semantic gBSD by modifying or removing segments or salient objects to satisfy user preferences.

However, this summarization is not sufficient to guarantee the generation of an adapted content that maximizes the user's QoE. In other words, SAF lacks of a quality metric that evaluates the subjective impact of the summarization in terms of perceived and semantic quality.

4.2.5 NinSuna : The Ninsna INtelligent Search framework for UNiversal multimedia Access

NinSuna is a fully-integrated platform for format-independent semantic-aware multimedia content adaptation and delivery engines based on semantic web technologies [133], [132], [134]. The objective behind its conception is to ensure UMA in streaming environments while addressing the issues of extensibility, scalability and interoperability in a multimedia delivery platform.

As argued by the authors, ensuring UMA in streaming environments is a challenging task especially when multimedia streams require temporal semantic adaptation, as in, shot/scene selection or video summarization. Actually, this type of adaptation makes use of semantic metadata to select or remove the shots and scenes in question, and entails cuts in the bitstreams. Besides the semantic interoperability issues between XML-based metadata standard, a temporal semantic adaptation introduces two major problems for compressed and synchronised multimedia streams (e.g., synchronized audio and video streams). Indeed, in compressed multimedia streams, dependencies exists between the frames (i.e., inter-coded frames are dependent on previous and/or following frames). Thus, to guarantee that the adapted stream can be decoded in a correct way, the cuts in the bitstream should only be performed at the random access points (i.e., frames independent of the previous ones). However, the random access points of synchronised multimedia streams do not coincide with each other. Therefore, it is crucial to keep the streams synchronisation intact whenever one of the streams undergoes the adaptation. Moreover, the position and frequency of the random access points depend on the coding format. To this end, a format-independent multimedia adaptation engine is desirable, given the growing amount in coding format.

The authors address these problems by developing the NinSuna plateform. To begin

with, NinSuna makes use of the Semantic Web technologies to enhance the interoperability among metadata standards for multimedia content. It uses Resource Description Framework (RDF) to describe the semantic relationships used during semantic adaptations. These semantics are explicitly represented by means of RDF triples, which are stored in the Resource Description Framework (RDF) repository. In addition, NinSuna uses the BSD tools (refer to Section 4.2.4), in order to provide semantic adaptation for the selection of scene of interest as well as for frame-rate reduction. It also provides the users with three services : retrieval, download and streaming. In order to request a particular multimedia bitstream, the users must send an Simple Protocol and RDF Query Language (SPARQL) query over the Resource Description Framework (RDF) repository. Afterwards, both the query and the description of the user's usage environment are sent to the ADTE. The latter generates the suitable adaptation plan and sends it to the adaptation and packaging engine. In NinSuna, the adaptation engine is built on AdaptationQoS, BSD and UED tools and relies on quality-based methods. It provides both coding format-independence and packaging format-independence techniques, thus enabling the adaptation of the streams in a format-independent way while keeping their synchronisation intact.

The main feature of this platform is to be independent of the metadata formats (e.g., MPEG-7, Dublin Core, etc.), coding formats (e.g., H.264/AVC, VC-1, etc.) and delivery formats (e.g., MP4, Ogg, etc.) of the multimedia content. This independency makes from NinSuna an extensible and interoperable platform since no changes in the general adaptation mechanism are required to support a new metadata, coding and delivery format. Furthermore, NinSuna relies on classical multi-attribute optimisation methods as in CAIN-21.

However, the authors do not discuss the completeness of the adaptation solution computed by the ADTE. Indeed, they do not describe how the ADTE deals with the problem of having none or several feasible adaptation plan satisfying the same user preference. Finally, NinSuna deals with semantic constraints proposed by the users by enabling temporal semantic adaptation (i.e., shot/scene selection or video summarization). However, NinSuna lacks of a quality metric that evaluates the subjective impact of this semantic adaptation in terms of perceived and semantic quality. Thereby, there is not a guarantee to generate the adapted bitstreams that maximizes the user's QoE.

4.3 Discussion and Positioning

In this section, we provide analysis and positioning of the previously discussed frameworks with respect to the thesis requirements (see Table 4.1-4.2). Accordingly, we study the feasibility and limitations of each framework in performing object-based adaptation while satisfying the thesis requirements.

TABLE 4.1 – Summary of video adaptation frameworks.

	Granularity level		**Adaptation operation**		**Adaptation approach**
	considered levels	*object*	*considered operations*	*removal*	
koMMa [63]	video, frame	No	format transcoding, scaling	No	technical-driven
CAIN [82]	video, frame, object	Yes	format transcoding, scaling, substitution	No	technical-driven + semantic-driven
DCAF [123]	video	No	format transcoding, scaling	No	technical-driven
DANAE [106], SAF [149]	video, scene, shot, frame	No	format transcoding, scaling, substitution, selection	No	technical-driven + semantic-driven
NinSuna [133]	video, scene, shot	No	format transcoding, removal, selection	Yes	technical-driven + semantic-driven
Thesis requirements	**video, scene, shot, frame, object**	**Yes**	**removal**	**Yes**	**semantic-driven**

TABLE 4.2 – Summary of video adaptation frameworks (follow).

	ADT method	**QoE-awareness**		**Adaptation focus**	**Conflicting constraints**	**Owner rights**
		SemQ	*PQ*			
koMMa [63]	knowledge-based	No	No	terminal	No	No
CAIN-21 [82]	hybrid	No	Yes	terminal + user	No	No
DCAF [123]	quality-based	No	No	terminal	No	No
DANAE [106], SAF [149]	hybrid	No	Yes	terminal + user	No	No
NinSuna [133]	quality-based	No	No	terminal + user	No	No
Thesis requirements	**quality-based**	**Yes**	**Yes**	**user**	**Yes**	**Yes**

4.3.0.1 koMMa :

In koMMa, the adaptation approach is not semantic-driven and only the format transcoding and scaling adaptation tools are provided. But, due to the modular nature of koMMa, new adaptation tools could be added. However, the representation format of the actions (i.e., adaptation operations) as it is defined, cannot express semantic constraints. As a consequence, neither the end-user constraints nor the owner constraints can be taken into consideration. In addition, even if the format issue were resolved, koMMa would still be unable to deal with conflicting constraints. Moreover, koMMa is a non-utility aware multimedia framework. The problem of finding the best adaptation solution among several ones to maximize the user experience is not considered. Thus, koMMa is not aware of the quality experienced by the end-user and therefore cannot ensure UME.

4.3.0.2 CAIN :

In contrast to koMMa, CAIN enables the end-user to express technical and semantic constraints. Therefore, the adaptation in CAIN is more user-centric. Moreover, while the adaptation decision-taking method is primarily knowledge-based as in koMMa, it is also combined with a quality-based method. This latter decides of the best solution among the available ones based on a ranking created by the VQM and PSNR quality metrics. However, these metrics are not suitable for measuring the semantic and perceptual quality after a semantic-driven adaptation involving spatio-temporal reduction (e.g., removal of objects within the frame, removal of shots, etc.). Regarding the semantic quality, both metrics are unable to measure the amount of information assimilated by the end-user after a semantic-driven adaptation. Regarding the perceptual quality, the fidelity metric PSNR disregards the characteristics of human visual perception [140], as discussed in Section 2.5.1. Furthermore, VQM is only developed to measure the perceptual quality after a technical- and perceptual-driven adaptations, but not after a semantic-driven adaptation. Indeed, VQM can only measure the perceptual effects of video impairments including blurring, jerky motion, global noise, block and color distortion, and combines them into a single value [100]. Finally, similar to koMMa, CAIN is not able to deal with conflicting constraints, which can occur when the users constraints violate the owner intellectual property rights.

4.3.0.3 DCAF :

As discussed in Section 4.2.3, DCAF can be seen as a special adaptation scenario in the case of CAIN. Indeed, as can be seen from Table 4.1-4.2, the characteristics of DCAF with respect to our evaluation criteria are generally similar to those of CAIN, with a few additional restrictions. The adaptation approach in DCAF is technical-driven and only the format transcoding and scaling adaptation tools (i.e., Web adaptation services) are provided. Moreover, the adaptation decision-taking method is quality-based, and the optimal service composition plan is selected in the graph based on user specified QoS criteria. However, these criteria do not consider the perceptual and semantic quality of the end-user. Therefore, DCAF lacks the necessary adaptation tools and quality metrics to fulfill the requirements of the scenarios proposed in this thesis. In addition, DCAF does not implement tools to generate semantic context and content descriptions. Finally, similar to koMMa and CAIN, DCAF is not able to deal with conflicting constraints.

4.3.0.4 DANAE & SAF :

As can be observed from Table 4.1-4.2, the DANAE project is the most mature among the other frameworks. It includes three different MPEG-21 standardized adaptation approaches. Besides SAF, these adaptation approaches are technical-driven and remedy to the diverse problems encountered by some of the aforementioned frameworks : incomplete solution, lack of context-awareness, inadequacy for dynamic adaptation, etc. However, the same shortcomings remain when it come to extend these approaches to be semantic-driven adaptation ones. Even SAF that provides semantic context and content descriptions to enable summarization, the adaptation decision-taking method is not appropriate for object-based adaptation methods. First, the limitation is related to the granularity of the adaptation operations, particularly the removal operation. Second, neither the semantic nor the perceptual quality are considered during the adaptation. Finally, the authors of SAF do not include the owner in the adaptation chain, or even address the problem of having conflicting semantic constraints.

4.3.0.5 NinSuna :

Despite being a fully-integrated platform for format-independent semantic-aware multimedia content adaptation and delivery engines in streaming environments, NinSuna cannot be used for applications requiring semantic object-based adaptation. In its current state, NinSuna can only perform temporal semantic adaptation by selecting fragments that match the user preferences/interests. But, since NinSuna is conceived to be extensible, scalable and interoperable, it could be extended to include spatial semantic adaptation. In this case, finer descriptions of the user preferences and the video content at the object level, would be required. Furthermore, the adaptation decision-taking method should be altered to handle simultaneously spatial and temporal adaptation. However, we were discourage to extend NinSuna for the purpose of this thesis, since the completeness of the adaptation solution computed by the ADTE is not discussed by the authors. Indeed, this is a central problem in our context because the quality of the resulting adapted video, can drastically change according to the selected adaptation plan. Finally, similar to SAF, the adaptation in NinSuna is not aware of the quality experienced by the end-user and, does not preserve the intellectual property rights of the owner.

As a conclusion of the analysis, none of the above frameworks can be adopted for the purpose of this thesis : their missing features for fulfilling the thesis requirements are far from being trivial. Indeed, as we will show in the remainder of this thesis, some of these features (e.g., resolving conflicting constraints), can only be provided by resolving difficult research problems. Introducing these original features has far-reaching consequences for the functioning of the video adaptation process, thus making the development of a new framework much more appropriate than extending the existing ones that we presented in this chapter.

To this end, we developed Personalized vIdeo Adaptation Framework (PIAF) [32], a quality-based semantic adaptation framework with spatio-temporal adaptation functionality. PIAF is a complete modular MPEG standard compliant framework that covers the whole process of semantic video adaptation. The semantic adaptation involves the removal adaptation operation at both spatial and temporal dimensions. Furthermore, PIAF enables both the user and the owner to express their preferences. In case of

conflicting constraints, the computation of a solution is resolved as an optimization problem.

4.4 Conclusion

In this chapter, we presented an analysis of the most known video adaptation frameworks that have been reported in the literature. To this end, we first defined the criteria of the analysis. Then, we individually described each framework and considered its pros and cons. Furthermore, we studied the feasibility and limitations of theses frameworks in performing object-based adaptation while satisfying the thesis requirements. As a conclusion of this analysis, we argued that none of these frameworks is adequate for the purpose of this thesis and we assessed the need of developing Personalized vIdeo Adaptation Framework (PIAF).

Part II

Formal Modeling of PIAF : Personalized vIdeo Adaptation Framework

Chapter 5

General Adaptation Framework

The purpose of this chapter is to provide a general introduction to the video personalization framework PIAF. Inspired by the adaptation systems described in the state of the art [85] [67], we thus present a simplified generic MPEG-based architecture for personalized video adaptation. The overall architecture follows the MPEG-21 vision, which is 'to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities' [53]. In the following, we discuss the functionality of each module and show how they can benefit from the tools provided by the MPEG-7 and MPEG-21 standards. We argue that using the MPEG-7 description tools in the MPEG-21 Multimedia Framework enables ***interoperable personalized semantic adaptation*** *of the multimedia content.*

5.1 MPEG-based Personalized Video Adaptation Architecture

The MPEG-based personalized video adaptation architecture is illustrated in 5.1. It is mainly composed of an MPEG-Description Generator and a Content Adaptation Engine. The former consists of two modules, one for describing the video content (Content Description) and one for describing the usage environment and the users constraints (Context Description). These modules provide the inputs to the Content Adaptation Engine, which in turn consists of two modules : Adaptation Decision Module and Adaptation Execution Module. The remainder of this section describes in detail the functionality of each module.

5.1.1 MPEG-Description Generator

As stated in the motivation chapter (Section 2.4.2), the availability of both content and context descriptions is a fundamental requirement for a content adaptation framework. Indeed, the quality of the content personalization mainly depends on whether the usage environment constraints and users preferences are properly expressed, and whether the description of the video content is rich and accurate enough in terms of content structure and semantics. The MPEG-7 and MPEG-21 standards provides a set of tools for the generation of these descriptions (refer to Section 3.2). In the following,

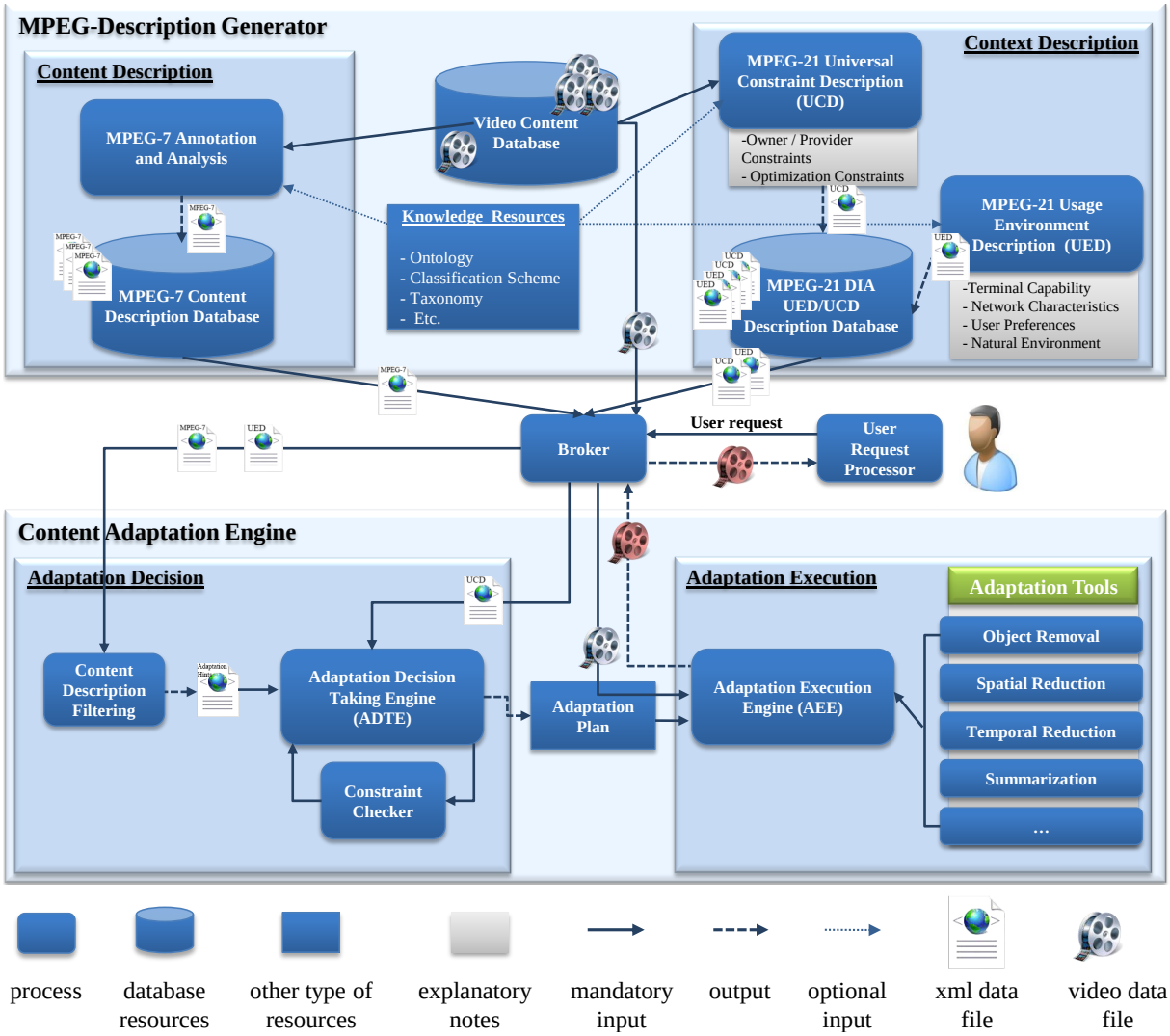


FIGURE 5.1 – A generic MPEG-based architecture for personalized video adaptation

we explain how the content and context description modules can benefit from these tools.

5.1.1.1 Content Description Module

The *Content Description Module* consists of an *MPEG-7 Annotation and Analysis* module, which takes a video as input and generates MPEG-7 XML descriptions of the content as output. This process implements the MPEG-7 Ds and DSs tools presented in Section 3.2.1. Depending on the type of descriptions needed for the adaptation application (e.g., for the content management or content description), this process either automatically extracts the content descriptions or requires human intervention (e.g., annotator, owner, etc.). For instance, technical adaptation applications require information about the media such as the storage format, the encoding format, the identification of the media, etc. This metadata which is represented by the MPEG-7 media description tools, can be automatically extracted. On the other hand, semantic and perceptual adaptation applications require structural (e.g., spatial, temporal and spatio-temporal relations) and semantic descriptions (e.g., free text, concepts) of the video elements. For this purpose, MPEG-7 provides content structure and semantic

description tools [9]. As this descriptions require human intervention, an annotation tool implementing the content structure and semantic description tools is needed. For interoperability reasons, the Content Description Module can benefit from the use of knowledge resources such as Ontologies, Taxonomies, Classification Schemes, etc. This guarantees the consistency of the vocabulary used by the module. Once these descriptions have been generated, they are sent to the *MPEG-7 Content Description Database* to be stored. One MPEG-7 content description should exist per video.

5.1.1.2 Context Description Module

The role of the *Context Description Module* is to collect information about the users (e.g., personal information such as age) and their usage context (i.e., users preferences, terminal, network and natural environment), and translate this information into an MPEG-based description format. As depicted in Figure 5.1, this module consists of an MPEG-21 Universal Environment Description (UED) and an MPEG-21 Universal Constraint Description (UCD) processes to generate these descriptions (see Section 3.2.2).

The MPEG-21 UED process is responsible for gathering information about the users characteristics and their usage context in order to model their profile. To this end, it implements the MPEG-21 UED tools. Depending on the requirements of the adaptation application, the user profile can be constructed manually (by explicit input) or can be inferred automatically (based on implicit input) [131]. For instance, in personalized semantic adaptation applications, users must explicitly specify their personal information (e.g., family name, age, etc.) and their preferences (e.g., semantic or technical constraints) through an interface. On the other hand, applications such as video recommendation systems automatically construct the user profiles by inferring information from their usage history (such as which type of videos the user viewed). It is worth noting that MPEG-21 UED includes usage history description tools for this purpose.

The MPEG-21 UCD process implements the MPEG-21 Universal Constraint Description (UCD) tool for describing users constraints for adaptation. These constraints must be always satisfied for any adaptation and they are enforced by the adaptation engine. In particular, the UCD tool supports the enforcement of the owner intellectual property rights by enabling content owners to formulate explicit constraints on how their videos can be manipulated. This process is implemented as a user interface. It takes as an input constraints descriptions and information about the video in question (e.g., video path, name, etc.), and generates a UCD XML file. Both UED and UCD files are stored in the *MPEG-21 DIA UED/UCD description database*.

Similar to the *Content Description Module*, the *Context Description Module* can also benefit from the use of knowledge resources to guarantee the consistency of the vocabulary used by its processes.

9. In the context of MPEG, the description tools comprise all of MPEG's predefined descriptors and description schemes

5.1.2 Content Adaptation Engine

The role of the Content Adaptation Engine is the following : given a video and a user, it tries to provide an adapted version of the video that satisfies the user and the owner constraints while maximizing their QoE. Upon a user request, the broker communicates with the MPEG-Description Generator to provide the Content Adaptation Engine with the necessary content and context descriptions. The content description is filtered according the context descriptions to locate the parts of the video that must to be adapted. Indeed, this filtering generates the information hints needed to drive the adaptation process. Two modules are involved in this process : 1) the adaptation decision taking engine that decides of the adaptation operations to be applied over the video, and 2) the adaptation execution engine that executes the selected operations.

5.1.2.1 Adaptation Decision Module

The *Adaptation Decision Module* takes UED, UCD and MPEG-7 content descriptions as input, and generates an adaptation plan as output. This adaptation decision can be automatically taken since all inputs descriptions have been generated according to a consistent vocabulary (as mentioned in the previous section).

The adaptation decision process starts with the *Content Description Filtering* process. The latter is responsible for filtering the MPEG-7 content descriptions according to the UED. The aim of this filtering is to identify the elements of the video concerned by the usage environment constraints, and to extract from the MPEG-7 content descriptions the information that the ADTE will need for the computation of an adaptation plan (such as the position and length of the shots to be removed). The result of this process is stored in an XML file.

Afterwards, the Adaptation Decision Taking Engine (ADTE) takes this file as input along with the UCD in order to find the appropriate adaptation operations over each adaptation entity (i.e., a video element), and computes the adaptation plan that maximizes the user's QoE and QoS. The task of the ADTE is challenging since the adaptation entity can be adapted along different dimensions (e.g., spatial, temporal). Moreover, more than one set of adaptation operations could meet the usage environment constraints, thus resulting in several feasible adaptation plans. The first challenge to complete this task is to objectively measure the quality of the adapted video, taking into consideration its multiple dimensions. The second challenge is to assess beforehand the quality of adapted content in order to choose the plan that will produce the best quality. As multiple constraints may be expressed not only by users but also by intellectual property right owners, the third challenge is to be able to resolve conflicting or overlapping constraints.

Regarding the development of an ADTE, MPEG-21 DIA [16] has standardized a quality-based AdaptationQoS tool. This tool addresses the problem of video adaptation to constraints imposed by terminals and/or networks for QoS management. In particular, it defines a Utility Function (UF) tool that describes for a set of constraints, the feasible adaptation operators and the resulting utilities. This tool efficiently solves the problem of adapting a video to the constraints related to terminal and network QoS management. However, the introduction of owner intellectual property rights and se-

mantic constraints in the adaptation process leads to a more complex optimization problem. The role of the constraint checker included in the adaptation decision module is to solve this problem. To this end, it must identify an optimal adaptation plan that has a high level of QoE while respecting the owner constraints.

The result of this process is an XML file representing the selected adaptation plan. This file can be stored internally, thus enabling the optimization of the delivery of adapted content by reusing previous adaptation plans [2]. Afterwards, this file is sent to the Adaptation Execution Engine (AEE) for execution.

5.1.2.2 Adaptation Execution Module

The *Adaptation Execution Module* takes the video and the adaptation plan as input, and produces an adapted video as output. The AEE is responsible for executing the adaptation plan on the video content. The AEE launches the appropriate adaptation tools over the video according to the adaptation operations of the plan. Indeed, an adaptation tool should exist for each adaptation operation. Figure 5.1 illustrates some of this tools such as *Object Removal* tool to remove an object from a frame (e.g., object inpainting) tool to reduce the spatial size (i.e., width×height) of the video content (e.g., spatial reduction of frames 640x480 to 320x240), etc.

5.1.3 Workflows

The architecture depicted in Figure 5.1 supports three main workflows : adding a new video, registering a new user, requesting a video. Based on the application scenario 1 presented in 2.2.1 and assuming that DownVid provides the user with a video content adaptation service from product placements, we illustrate in the following each workflow. Moreover, at the end of each workflow, we refer the readers to the chapters that deal with it.

Adding a new video : DownVid wants to add the 'Dark Tree' movie to its video database. First, the video is sent to the *Content Description Module* to locate the product placements objects, extract their spatio-temporal information and annotate them according to a controlled vocabulary for brands names. The resulting *MPEG-7 XML Descriptions* file is stored in the MPEG-7 content description database. Then, the video is sent to the MPEG-21 UCD interface enabling the owner to express her adaptation constraints, such as for instance 'the video cannot be shortened more than 10%'. The output UCD is stored in the *MPEG-21 DIA UED/UCD Description Database*.
In Chapter 7, we present a formal model of the spatial, temporal, and semantic information generated by video analysis algorithms. With respect to the implementation of this model, Chapter 11 presents our Semantic Video Content Annotation Tool (SVCAT), which assists annotators in generating video annotations according to the proposed video model. For a step-by-step example of a video content annotation using SVCAT, we refer the reader to Annex B.

Registering a new user : Pascal wants to be a member of DownVid. The creation of a new *User Profile* is done through a web browser interface. The fields of the

registration form are divided into two groups : those regarding the user's personal information (e.g., family name, age, citizenship, etc.) and authentication fields (e.g., log in, password, email address, etc.), and those regarding the user preferences presented as a checklist of adaptation services provided by DownVid (e.g., video content adaptation service from product placements). Let us assume that Pascal wants to benefit from the adaptation service and has successfully registered to DownVid. Therefore, a UED file is generated and stored in the *MPEG-21 DIA UED/UCD Description Database*. Besides the owner constraints, DownVid as a provider can also impose constraints, which can be inferred from the information in the user profile. In order to enforce Turkey's legislation, DownVid can define constraints as a form of logical sentences : '*Adapt(alcohol and tobacco advertisement)* **if** *citizenship(end-user)= Turkey*' [10]. The inferred constraints are then expressed in a separate UCD system file.
The concept of user semantic constraint in the user profile is formally defined in Chapter 8, while its representation using the MPEG-7 and MPEG-21 standards is detailed in Chapter 12. An exemplary instance of a user profile including its semantic constraints is presented in Annex C.

Requesting a video : Pascal authenticates to DownVid and requests to watch 'Dark Tree'. DownVid identifies him and retrieves his UED. According to Pascal's preferences, the video should be adapted. The system queries the video database and finds the requested video as well as its MPEG-7 XML Descriptions and UCD XML files. To begin, the MPEG-7 descriptions file is filtered against the UED to locate the parts of the video that must undergo the adaptation process. Once this has been done, these adaptation information hints are sent to the ADTE to assist it in choosing an adaptation plan. Let us assume that DownVid has implemented adaptation tools for two adaptation operations : object removal and frame sequence removal. Thus, the ADTE must decide of the best combination of adaptation operations taking into account the owner and DownVid constraints expressed in the UCD files while satisfying Pascal's preferences and maximizing his QoE. This leads to solve the adaptation decision as an optimization problem. Once the adaptation plan is found, it is sent along with the video to the Adaptation Execution Engine (AEE) for execution. Finally, the adapted video is sent to Pascal.
A formal modelling of the semantic constraint instantiation process including the adaptation information hints are presented in Chapter 8. Chapter 12 describes in detail the implementation of this process along with representation of the information hints. Given this information, the process of computing the best combination of adaptation operations taking into account both owner and user constraints, is formally described in Chapter 9 and Chapter 10. The evaluation of the performance and accuracy of the ADTE process are presented in Chapter 13.

10. We insist on the fact that this type of constraint is used to enforce the legislation of a country, but not to limit the freedom of a user.

Chapter 6

Preliminaries

In this chapter, we review the fundamental properties of the concepts used throughout this thesis and introduce the formalisms that we have adopted for representing them.

6.1 Set

We always denote a finite set by explicitly specifying all of its elements between the curly brackets {}, such that the set is represented by a capital letter, e.g., $X = \{1,\ 2,\ 3,\ 4,\ 5,\ 6\}$. The cardinality of set X is denoted by $|X|$. If the elements in a set S with $|S| = n$ are not set-valued, then they are represented by a lower case and are indexed from 1 to n, e.g., $S = \{s_1,\ s_2,\ s_3,\ s_4,\ s_5\}$. If the elements in a set follow an obvious pattern, we define the set in a compact way by enumerating the first and last element. For instance, the set X and S will be represented by $X = \{1,\ \ldots,\ 6\}$ and $S = \{s_1,\ \ldots, s_5\}$, respectively. We express an arbitrary element of a set by writing $s \in S$, and a specific element by indicating its index $s_3 \in S$.
In cases where the elements of a set possess a particular property p (or set of properties), then we define the set as follows : $S = \{s_i \mid p(s_i) = true\ for\ all\ i\ in\ a\ domain\ D \subset \mathbb{N}\}$. We denote by $\mathcal{P}(S)$ the power set (or powerset) of the set S, which is the set of all subsets of the set S. For instance, the power set of $S = \{s_1,\ s_2\}$ is $\mathcal{P}(S) = \{\emptyset,\ \{s_1\},\ \{s_2\},\ \{s_1,\ s_2\}\}$.

6.2 Function

We define a function by first giving a textual description of the domain, codomain and its assignment, followed by a formal definition. For example, let f be a function whose domain and codomain are the finite sets X and Y, respectively. For each $x \in X$, an assignment expressed as $f(x) = y$ means that f assigns $x \in X$ to $y \in Y$. The function f is formally defined as follows :

$$\begin{array}{rccl} f & : & X & \longrightarrow & Y \\ & & x & \longmapsto & f(x) = y \end{array}$$

For complex functions, we always complement the formal definition by an example.

6.3 Sequence

Let S be a finite set of non set-valued elements. We consider a sequence of elements of S, also called a sequence in S, as a function whose domain $K = \{n \in \mathbb{N} \mid n_0 \leq n \leq l \text{ such that } n_0 \geq 1 \text{ and } l \leq n_0 + |S|\}$ is a countable and totally ordered subset of the natural numbers $\mathbb{N}$, and its codomain is S. We denote a sequence in S, the function f given by :

$$\begin{array}{rccl} f & : & K \subset \mathbb{N} & \longrightarrow & S \\ & & n & \longmapsto & f(n) = s_n \end{array}$$

For convenience, we always denote by f the function for a sequence in S, and by I_s^v instead of K the domain of f for a sequence $v = (s_n)$ of elements in S. For the sake of simplicity, we denote a sequence by $(s_n)_{n \in I_s^v}$ or $(s_n)_{n=n_0}^{l} = (s_{n_0},\ s_{n_0+1},\ \ldots,\ s_l)$ instead of $f(n)$, such that s_n is called the n^{th} term of the sequence, s_{n_0} is the first element of the sequence whereas n_0 is bounded below with 1 and s_l is its last element whereas l is bounded above with $n_0 + |S|$.

For example, let $S = \{s_1,\ s_2,\ s_3,\ s_4,\ s_5\}$ and $I_s^v = \{n \in \mathbb{N} \mid 1 \leq n \leq 5\}$ be two finite sets. We denote by $v = (s_n)_{n=1}^{5} = (s_1,\ s_2,\ \ldots,\ s_5)$ the sequence in S.

Since a sequence is defined as a function, thus all concepts defined for functions (bounds, monotonicity , ...) also apply to sequences. In the following, we just review the definitions needed for the purpose of this thesis.

6.3.0.1 - One-to-one Sequences

A sequence $v = (s_n)_{n \in I_s^v}$ is said to be one-to-one if it is a sequence of distinct terms : $\forall i,\ j\ \in I_s^v,\ i \neq j\ \Longrightarrow s_i \neq s_j$

6.3.0.2 - Strictly Increasing Sequences

Let the set S be totally ordered with respect to $<$. A sequence $v = (s_n)_{n \in I_s^v}$ in S is said to be strictly increasing if each term is greater than the previous one : $\forall n \in I_s^v$: $s_n < s_{n+1}$.

6.3.0.3 - Equality of Sequences

Let two sets X and Y be totally ordered with respect to $<$, and f and g be respectively two sequences in X and Y :

- $f = (x_1,\ \ldots,\ x_n)$
- $g = (y_1,\ \ldots,\ y_m)$

Then f is equal to g iff :

- $m = n$
- $\forall i : 1 \leq i \leq n : x_i = y_i$

6.4 Subsequence of a Sequence

Let X be a finite set and $f = (x_n)_{n \in I_x^f}$ be a sequence in X. A sequence $g = (y_n)_{n \in I_y^g}$ *s. t.* $I_y^g = \{n \in \mathbb{N} \mid n_0 \leq n \leq l\}$ is a subsequence of $f = (x_n)_{n \in I_x^f}$ iff :

1. $g = (y_n)_{n \in I_y^g}$ is a sequence of elements in X, and
2. there is a strictly increasing function $q : I_y^g \longrightarrow I_x^f$ such that : $y_n = x_{q(n)}$ $for\ all\ n \in I_y^g$

That is, the subsequence $g = (y_n)_{n \in I_y^g}$ is obtained by choosing terms from the original sequence $f = (x_n)_{n \in I_x^f}$, without altering the order of the terms, through the mapping function q, which determines the indices used to pick out the subsequence.

For example, let $X = \{a,\ b,\ c,\ d,\ e\}$ be a finite set and $f = (a,\ b,\ c,\ d,\ e)\ s.\ t.\ I_x^f = \{1, 2, 3, 4, 5\}$ be a sequence in X. A sequence $g = (a,\ d)\ s.\ t.\ I_y^g = \{1, 2\}$ is a subsequence of $f = (x_n)_{n \in I_x^f}$ because :

1. $g = (y_n)_{n \in I_y^g}$ is a sequence of elements in X, that is $\{a,\ d\} \subset \{a,\ b,\ c,\ d,\ e\}$, and
2. there is a strictly increasing function $q : I_y^g \longrightarrow I_x^f$ with $q(n) = n^2$ such that : $y_n = x_{q(n)}$ $for\ all\ n \in I_y^g$

During this thesis, we denote by q_v the strictly increasing function for a subsequence v.

Chapter 7

Video Model

In this chapter, we present a formal model of the spatial, temporal, and semantic information generated by video analysis algorithms. The formalization is generic and independent of the method used by the analysis algorithms. To begin with, based on the definition that a video is a sequence of ordered image frames, we formally model the frame as an image. To this end, we review some definitions related to the image segmentation process and adopt them to formalize the spatial information of a frame. Furthermore, we present a generic formal model for the temporal structure of a video, which is independent of the shot and scene segmentation algorithms. Finally, given both spatial and temporal formal definitions, we formally define the information related to the object (semantic, spatial and temporal properties).

7.1 Spatial Structure of Video Data

The spatial structural elements of a video are pixel, frame and region. Since these elements are well-defined in the image processing domain, we quickly recall some definitions from [7], [36], [46] and refer the reader to these papers for motivation. Annex A also recalls some fundamental properties of the pixel adopted from [36].

7.1.1 Pixel

Definition 1. Pixel : As defined in [7], a pixel p (abbreviation for "picture element"), is a small dot of colored lights. It is the basic unit of programmable color on a computer display or in a digital image.

Pixel Value : As defined in [36], each of the pixels representing a digital image has a pixel value which represents its color.

Example 1. In the simplest case of binary images, the pixels have only two possible values, black and white. Numerically, the two values are often 0 for black, and 1 for white.

7.1.2 Frame

Definition 2. Frame : A frame f (also called image frame) is an image defined as a $m \times n$ matrix $f = [a_{i,j}]_{m \times n}$, such that the indices specify the position of a pixel at the i^{th} row and j^{th} column in the frame and the element $a_{i,j}$ as the value of the pixel.

Notation 1. Set of all the pixels of a Frame : We denote P^f the set of all the pixels of a frame f.

Definition 3. Frame Resolution : The frame resolution denoted by $size_f$ refers to the number of pixels that constitute the frame f. It is defined as $size_f = m \times n$.

7.1.3 Region

A region inside a frame is a set of connected pixels, which are homogeneous with respect to some characteristics (e.g., gray tone or texture) [46]. The definition of the homogeneity condition depends on the frame segmentation (i.e., image segmentation) algorithm used in the application.

Let us assume the existence of a frame segmentation algorithm consisting of a function, which determines if two pixels p and q have the same characteristic, or not. We note $p\ hm_c\ q$ if the pixels having the same characteristics c, and by $p\ \neg hm_c\ q$ otherwise.

Definition 4. c-Homogeneity : Given a set of pixels $S^f \subseteq P^f$ of a frame f, the predicate $c - Homogeneity$ is defined as follows :

$$\begin{aligned} c - Homogeneity(S^f) = &\quad true &&\iff \quad p\ hm_c\ q,\ \forall\ p,\ q \in S^f \\ c - Homogeneity(S^f) = &\quad false &&\iff \quad \exists\ p,\ q \in S^f\ s.\ t.\ p\ \neg hm_c\ q \end{aligned}$$

Figure 7.1 illustrates a frame segmentation technique described in [95], which is based on color characteristics. The original and the segmented frame are illustrated on the left and the right, respectively. For instance, the flower consists of four different connected regions.

FIGURE 7.1 – Color-based frame segmentation, adapted from [95]

Accordingly, the set of regions of a frame is the result of the frame segmentation.

For a formal definition of the frame segmentation, we refer the readers to the following reference [38]. Thus, we formally define the concept of region in a frame as follows :

Definition 5. Region : Let $r^f \subseteq P^f$ be a connected component in a frame f satisfying the predicate $c - Homogeneity(r^f)$. Then, r^f is said to be a region.

Cardinality of r^f : We denote by $|r^f|$ the cardinality of the finite set r^f. It corresponds to the number of pixels constituting the region r^f.

Set of Regions in a Frame : Let f be a frame and a frame segmentation algorithm. Then, the set of regions $R^f = \{r_1^f,\ r_2^f,\ \ldots,\ r_n^f\}$ resulting from the frame segmentation is said to be the set of regions in the frame f.

7.2 Temporal Structure of Video Data

7.2.1 Video

A video can be seen as a sequence of image frames (i.e., frames), which convey a rich semantic presentation through the synchronized display of the frames over a period of time. Thus, the fundamental units of the video are single frames [122] as they are atomic on the time axis [37]. All the frames of the same video have the same frame resolution (see Definition 3).

In the remainder of this chapter, we will use the properties and notations of the concepts presented in the preliminaries. We use these concepts without describing them in detail, and refer the reader to Chapter 6 for more information.

Set of all frames in a video : We denote $\mathcal{F}^v = \{f_1,\ f_2,\ \ldots,\ f_n\}$ the set of all the frames of a given video v.

In the rest of the thesis, we omit for the sake of simplification the mention of the video v since all the concepts that we define are related to a single video. For instance, we use the simplified notation $\mathcal{F}$ instead of $\mathcal{F}^v$.

Binary relation $<$ on $\mathcal{F}$: We equip the set $\mathcal{F}$ with a binary relation $<$. This binary relation defines a total order on $\mathcal{F}$ and corresponds to the order in which the frames are meant to be displayed.

Definition 6. Video : Let $(f_n)_{n=1}^{|\mathcal{F}|}$ be a one-to-one and strictly increasing sequence of elements of $\mathcal{F}$, the domain of which is $\{n \in \mathbb{N} \mid 1 \leq n \leq |\mathcal{F}|\}$. Then, the sequence $v = (f_n)_{n \in I_f^v}$ in $\mathcal{F}$ where $I_f^v = \{n \in \mathbb{N} \mid 1 \leq n \leq |\mathcal{F}|\}$ is said to be a video.

As stated in Section 6.3, we note I_f^v the domain for the sequence $v = (f_n)$ of elements f in $\mathcal{F}$. This notation is important as it holds information about the sequence and the type of its elements.

Example 2. Given the set of frames $\mathcal{F}$ such that $|\mathcal{F}| = 450$ and $I_f^v = \{1,\ 2,\ \ldots,\ 450\}$,

the video is thus defined as $v = (f_n)_{n \in \{1,\ 2,\ ...,\ 450\}} = (f_n)_{n=1}^{450} = (f_1,\ f_2,\ \ldots,\ f_{449},\ f_{450})$.

Definition 7. Length of a Video : The length of a video sequence denoted by $length_v$ is the number of frames constituting the video. It is defined as $length_v = |I_f^v|$ such that $|I_f^v|$ is the cardinality of the domain of v.

Since the domain of the video sequence I_f^v is a countable and totally ordered set, we adopt the following definition of video intervals given by Pradhan et al. in [102]. Furthermore, we review the intersection properties for both video intervals and set of video intervals, which we will need in Section 9.2.3 of Chapter 9.

Definition 8. Video Intervals : Given the domain of the video sequence I_f^v a countable and totally ordered set, a video interval $D[a, b]$ over I_f^v is the set of indices $\{c \in I_f^v \mid a \leq c \leq b\}$; a and b of $D[a, b]$ are the indices of the so-called starting frame and ending frame, denoted by $start(D)$ and $end(D)$, respectively.

For instance, back to Example 2, $[1, 450] = I_f^v$ is the interval of the full video.

Notation 2. Set of all video intervals over I_f^v : We note $\mathcal{D}(I_f^v)$ the set of all the video intervals over I_f^v

Definition 9. Length of a Video Interval : The length $length_D$ of a video interval $D[a, b]$ corresponds to the number of frames in the interval. It is defined as $length_D = (b - a) + 1$.

Property 1. Intersection of Video Intervals : Given two video intervals $D1$ and $D2$ over $\mathcal{D}(I_f^v)$, the two video intervals D_1 and D_2 are said to be intersecting iff $\exists c,\ c \in D_1\ and\ c \in D_2$, that is, $max(start(D_1), start(D_2)) \leq min(end(D_1), end(D_2))$. The operation interval intersection $\odot$ on any two arbitrary video intervals yields a single video interval D as follows :

$$D_1 \odot D_2 = \begin{cases} D[a, b] & ,\ if\ D_1\ and\ D_2\ are\ intersecting \\ \\ \emptyset & ,\ otherwise \end{cases} \tag{7.1}$$

such that $a = max(start(D_1), start(D_2))$ and $b = min(end(D_1), end(D_2))$.

Example 3. Thus, supposing there are two video intervals $D_1[10, 30]$ and $D_2[15, 40]$ then $D_1 \odot D_2 = D[15, 30]$. However, for $D_1[10, 30]$ and $D_2[35, 40]$, $D_1 \odot D_2$ does not produce any interval ($D_1 \odot D_2 = \emptyset$).

Property 2. Intersection of Sets of Video Intervals : Given two sets of video intervals X and Y, the interval set intersection operation $\bigodot$ on the two sets returns a set of video intervals constituting of the pairwise interval intersection $\odot$ between the elements of the two input sets.

$$X \bigodot Y = \{x \odot y \mid x \in X,\ y \in Y,\ x\ and\ y\ intersect\}$$

Example 4. Given two sets of video intervals $A = \{[2, 7],\ [41, 95]\}$ and $B = \{[5, 55],\ [74, 100]\}$, then the intersection of the sets is :
$A \bigodot B = \{[2, 7] \odot [5, 55],\ [2, 7] \odot [74, 100],\ [41, 95] \odot [5, 55],\ [41, 95] \odot [74, 100]\}$
that is $A \bigodot B = \{[5, 7],\ [41, 55],\ [74, 95]\}$ by applying Property 1.

7.2.2 Frame Sequence

Definition 10. Frame Sequence : A frame sequence fs is a finite sequence of consecutive frames in a video v. It is defined as $fs = (f_n)_{n \in I_f^{fs}}$ as a one-to-one and strictly increasing sequence of elements of $\mathcal{F}$, the domain of which is $I_f^{fs} = \{n \in \mathbb{N} \mid n_0 \leq n \leq l\}$.

If $n_0 = 1$ *and* $l = |\mathcal{F}|$ then $I_f^{fs} = I_f^v$ and the frame sequence is equal to the video.

Property 3. : A frame sequence $fs = (f_n)_{n \in I_f^{fs}}$ is a subsequence of the video $v = (f_n)_{n \in I_f^v}$ as both conditions are verified :

1. fs is a sequence of elements in $\mathcal{F}$, and
2. there is a strictly increasing function $q_{fs} : I_f^{fs} \longrightarrow I_f^v$ with $q_{fs}(n) = n$ such that $(f_n)_{n \in I_f^v} = (f_{q_{fs}(n)})_{n \in I_f^{fs}}$.

Example 5. Given the video $v = (f_1,\ f_2,\ \ldots,\ f_{449},\ f_{450})$ and the set $I_f^{fs} = \{n \in \mathbb{N} \mid 8 \leq n \leq 43\}$, thus $fs = (f_n)_{n \in \{8,\ 9,\ ...,\ 43\}} = (f_n)_8^{43} = (f_8,\ f_9,\ \ldots,\ f_{42},\ f_{43})$ is a frame sequence in the video v.

Property 4. : The domain of every subsequence of the video is a video interval.

7.2.3 Shot

As mentioned in section 2.4.1, a shot is a continuous sequence of frames captured from the same camera direction and view angle. The notion of *frame continuity* for a video shot depends on the shot detection algorithm used in the application. For instance, Albanese et al. in [5] define a frame sequence as a 'block' and formalize the notion of *frame continuity* for a video block. Accordingly, they introduce the predicate *isShot*, which takes a video block as an input and returns true or false as an output. In the following, we explain how we adapt these definitions to our video model and give our definition of a shot.

Let us assume the existence of a shot detection algorithm, which is capable of detecting whether two consecutive frames in a frame sequence are captured from the same camera direction and view angle, or not. We denote by $f_n \rhd_{sh} f_{n+1}$ if the frames respect the aforementioned condition, and by $f_n \not\rhd_{sh} f_{n+1}$ otherwise.

Definition 11. Continuity : Given a frame sequence $fs = (f_n)_{n=n_0}^{l}$, the predicate *Continuity* of frame sequence being captured from the same camera direction and view angle, is defined as follows :

$$\begin{aligned} Continuity(fs) = \ & true \iff f_n \rhd_{sh} f_{n+1},\ \forall\ n \in [n_0, l-1] \\ Continuity(fs) = \ & false \iff \exists\ n \in [n_0, l-1]\ s.\ t.\ f_n \not\rhd_{sh} f_{n+1} \end{aligned}$$

Moreover, let us assume the existence of a predicate called $isShot()$, which takes a video frame sequence as an input and returns true or false as an output. The predicate $isShot()$ must satisfy the following axiom : $isShot(fs) = true \iff Continuity(fs) =$

true.

Definition 12. Shot : Let $fs = (f_n)_{n=n_0}^{l}$ be a frame sequence satisfying the predicate $isShot(fs) = true$. Then, the sequence $sh = fs = (f_n)_{n \in I_f^{sh}}$ whereas $I_f^{sh} = \{n \in \mathbb{N} \mid n_0 \leq n \leq l\}$ is said to be a shot.

Definition 13. Length of a Shot : The length of a shot sequence denoted by $length_{sh}$ is the number of frames constituting the shot. It is defined as $length_{sh} = |I_f^{sh}|$ such that $|I_f^{sh}|$ is the cardinality of the domain of sh.

Definition 14. Size of a Shot : The size of a shot denoted by $size_{sh}$ refers to the number of pixels of frames within the shot. It is defined as the length of the shot in frames multiplied by the frame resolution : $size_{sh} = length_{sh} \times size_f$.

Set of all the shots in a video : We denote by $\mathcal{SH}$ the set of all the shots of a video v, such that the union of the sets of elements for all the shots is equal to the set $\mathcal{F}$, and their intersection is the empty set. Let $I_f^{sh_i}$ be the domain of a shot sh_i and I_f^v the domain of v, we formally define the set $\mathcal{SH}$ as follows :

$$\begin{aligned}\mathcal{SH} = \; & \{\, sh_i = (f_n)_{n \in I_f^{sh_i}} \mid min(I_f^{sh_1}) = 1,\ max(I_f^{sh_l}) = |\mathcal{F}|, \\ & min(I_f^{sh_{i+1}}) = max(I_f^{sh_i}) + 1,\ \bigcup_{i=1}^{|\mathcal{SH}|} I_f^{sh_i} = I_f^v \ and, \\ & \bigcap_{i=1}^{|\mathcal{SH}|} I_f^{sh_i} = \emptyset \ for\ all\ i = 1,\ 2,\ \ldots,\ l = |\mathcal{SH}| \,\}\end{aligned}$$

Binary relation $<$ on $\mathcal{SH}$: We equip the set $\mathcal{SH}$ with a binary relation $<$. This binary relation defines a total order on $\mathcal{SH}$ and corresponds to the order in which the shots are meant to be displayed.

Example 6. Going back to example 2, the set of shots for the video $v = (f_1,\ f_2,\ \ldots,\ f_{450})$ is defined as $\mathcal{SH} = \{sh_1 = (f_1,\ f_2,\ \ldots,\ f_{45}),\ sh_2 = (f_{46},\ f_{47},\ \ldots,\ f_{138}),\ \ldots,\ sh_6 = (f_{390},\ f_{391},\ \ldots,\ f_{450})\}$, so $|\mathcal{SH}| = 6$.

7.2.4 Shot Sequence

Sequence in $\mathcal{SH}$: Let $(sh_n)_{n \in I_{sh}}$ be a one-to-one and strictly increasing sequence in $\mathcal{SH}$, the domain of which is $I_{sh} = \{1,\ 2,\ \ldots,\ |\mathcal{SH}|\}$.

Definition 15. Shot Sequence : A shot sequence $shseq$ is a finite sequence of consecutive shots in a video v. Formally, $shseq = (sh_n)_{n \in I_{sh}^{shseq}}$ is a one-to-one and strictly increasing sequence of elements of $\mathcal{SH}$, the domain of which is $I_{sh}^{shseq} = \{n \in \mathbb{N} \mid n_0 \leq n \leq l\}$.

Example 7. Going back to example 6, given the set of shots $\mathcal{SH}$ and the sequence $(sh_n)_{n \in I_{sh}}$ in $\mathcal{SH}$ such that $I_{sh} = \{1,\ 2,\ \ldots,\ 6\}$, then $shseq = (sh_n)_{n \in \{2,\ 3,\ 4\}} = (sh_n)_2^4 = (sh_2,\ sh_3,\ sh_4)$ is a shot sequence.

7.2.5 Scene

As mentioned in section 2.4.1, a scene is a sequence of shots that are semantically related and narrate the same events. The definition of the notion of *semantically related shots* for a video scene depends on the scene segmentation algorithm used in the application. Similar to the definition of a shot, we formalize the notion of *semantically related shots* as follows :

Let us assume the existence of a scene detection algorithm, which is capable of detecting whether two consecutive shots in a shot sequence are semantically related and narrating the same events, or not. We denote by $sh_n \sim_{sc} sh_{n+1}$ if the shots respect the aforementioned condition, and by $sh_n \nsim_{sc} sh_{n+1}$ otherwise.

Definition 16. SemSimilarity : Given a shot sequence $shseq = (sh_n)_{n=n_0}^{l}$, the predicate $SemSimilarity$ is defined as follows :

$$\begin{aligned} SemSimilarity(shseq) = \ & true \iff sh_n \sim_{sc} sh_{n+1},\ \forall\ n \in [n_0, l-1] \\ SemSimilarity(shseq) = \ & false \iff \exists\ n \in [n_0, l-1]\ s.\ t.\ sh_n \nsim_{sc} sh_{n+1} \end{aligned}$$

Moreover, let us assume the existence of a predicate called $isScene()$, which takes a video shot sequence as an input and returns true or false as an output. The predicate $isScene()$ must satisfy the following : $isScene(shseq) = true \iff SemSimilarity(shseq) = true$.

Definition 17. Scene : Let $shseq = (sh_n)_{n=n_0}^{l}$ be a shot sequence satisfying the predicate $isScene(shseq) = true$. Then, the sequence $sc = shseq = (f_n)_{n \in I_{sh}^{sc}}$ whereas $I_{sh}^{sc} = \{n \in \mathbb{N} \mid n_0 \leq n \leq l\}$ is said to be a scene.

The set of all scenes in a video : Let $\mathcal{SC}$ denote the set of all scenes for a given video v, such that the union of the sets of elements for all the scenes is equal to the set $\mathcal{SC}$, and their intersection is the empty set. Let $I_{sh}^{sc_i}$ be the domain of a scene sc_i and $I_{sh} = \{1,\ 2,\ \ldots,\ |\mathcal{SH}|\}$, we formally define the set $\mathcal{SC}$ as follows :

$$\begin{aligned} \mathcal{SC} = \ & \{\ sc_i = (sh_n)_{n \in I_{sh}^{sc_i}} \mid min(I_{sh}^{sc_1}) = 1,\ max(I_{sh}^{sc_l}) = |\mathcal{SH}|, \\ & min(I_{sh}^{sc_{i+1}}) = max(I_{sh}^{sc_i}) + 1,\ \bigcup_{i=1}^{|\mathcal{SC}|} I_{sh}^{sc_i} = I_{sh}\ and, \\ & \bigcap_{i=1}^{|\mathcal{SC}|} I_{sh}^{sc_i} = \emptyset\ for\ all\ i = 1,\ 2,\ \ldots,\ l = |\mathcal{SC}|\ \} \end{aligned}$$

Example 8. Going back to example 6, the set of all the scenes for the video v is defined as $\mathcal{SC} = \{sc_1 = (sh_1,\ sh_2),\ sc_2 = (sh_3),\ sc_3 = (sh_4,\ sh_5,\ sh_6)\}$, so $|\mathcal{SC}| = 3$.

Definition 18. Length of a Scene : The length of a scene sequence denoted by $length_{sc}$ is the number of frames constituting the scene. It is the sum of the length of the shot elements in sc given by : $length_{sc} = \sum_{i \in I_{sh}^{sc}} length_{sh_i}$ such that $|I_{sh}^{sc}|$ is the cardinality of the domain of sc.

7.3 Object Representation

7.3.1 Object

As defined in Section 7.1.3, the result of a frame segmentation is a set of regions R_f. Based on this definition, some elements of R_f can be grouped to form a geometrical representation that corresponds to an object, which can be recognized without ambiguity as a real-world object.
We assume the existence of a process able to perform such grouping (e.g., manual selection, recognition from existing object patterns, etc.). We refer to a grouped set of regions for a specific object as a spatial object, which we formally define as follows :

Definition 19. Spatial Object : Let f be a segmented frame with $R^f = \{r_1^f, r_2^f, \ldots, r_n^f\}$ being its set of regions. Then, a subset of regions $so^f \subseteq R^f$ corresponding to a real world object is said to be a spatial object.

Definition 20. Size of a Spatial Object : The size of a spatial object denoted by $size_{so^f}$ is defined as the sum of the cardinalities of its regions.

Figure 7.2 depicts the spatial object so^f of a flower, which is composed of four regions.

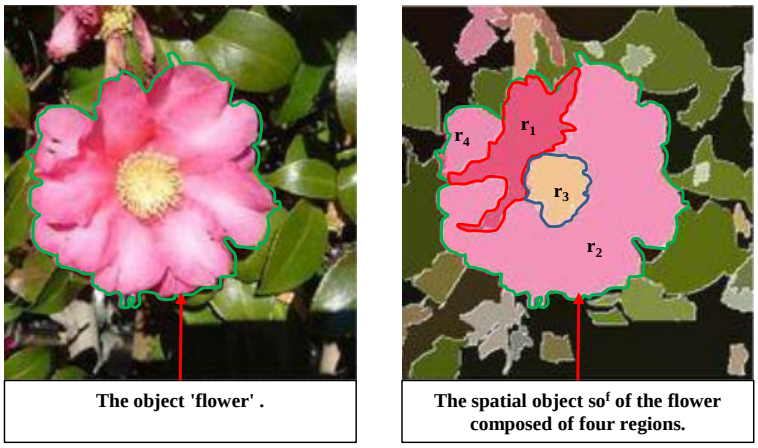


FIGURE 7.2 – Spatial object so^f of a flower, adapted from [95].

Given a spatial object so^f in a frame f, the process of finding all the similar spatial objects in each frame of the video is called object tracking. The definition of the similarity depends on the object tracking algorithm used in the application. We assume the existence of an object tracking algorithm, which takes segmented video frames and a spatial object as an input, and returns the sets of similar spatial objects in different frames as an output.

Accordingly, we define the predicate $Contains()$, which takes a frame from the video and a spatial object as an input and returns true if a spatial object similar to the input is found in the frame, or false otherwise. Let f_i be a frame of a video v, and so^f a spatial object identified in a frame f, the predicate $Contains(f_i, so^f)$ is true iff it exists a spatial object so^{f_i} in the frame f_i such that so^{f_i} is similar to so^f.

Set of the Spatial Objects in a Video : Given a video $v = (f_n)_{n \in I_f^v}$ and a

spatial object so^f in a frame f to be tracked, the set of all spatial objects related to so^f is given by : $SO = \{so^{f_i} \mid \forall\ i \in I_f^v,\ Contains(f_i, so^f) = \ true\}$.

Furthermore, we use MPEG-7 compliant classification schemes to semantically describe the object. Indeed, as mentioned in Chapter 5, PIAF is an MPEG standard compliant framework. Thus, an MPEG-7 metadata format is required, in order to enable the integration and exploitation of the generated metadata by the various modules of PIAF. In the following, we recall the definitions of Term and Classification Scheme presented in Section 3.2.1.4, and then give the formal definition of an object.

Term : A term t is a description associated with an object representing an entity (e.g., car). A term is defined as a 3-tuple $t = \langle id_t,\ name_t,\ def_t \rangle$, where id_t, $name_t$ and def_t correspond respectively to the identifier, name, and definition of the term. The identifier uniquely identifies the term, the name may be displayed to a user or used as a search term in a target database, and the definition describes the meaning of the term.

Classification Scheme : A classification scheme CS is a set of standard terms for a specific domain d. A classification scheme is defined as a 2-tuple $CS_d = \langle T_d,\ term - relation \rangle$, where T_d is the set of terms related to a domain d and $term - relation$ is the relation between terms.

Definition 21. Object : An object o is defined as a 2-tuple $o = \langle T_o,\ SO \rangle$, such that T_o is the set of terms associated with the object to describe its semantics and SO the set of all spatial objects related to o in all the frames of the video.

Notation 3. Set of all the Objects in the Video : We denoted by $\mathcal{O}$ the set of all the different objects in a given video.

7.3.2 Object Frame

An object frame is a frame containing a spatial object belonging to an object. Thus, we define the predicate $isOf()$, which takes a frame and an object as input and returns true if a spatial object belonging to the input object is found in the frame, or false otherwise. The predicate $isOf()$ is formally defined as :
$\forall\ o = \langle T_o,\ SO \rangle \in \mathcal{O}\ and\ f \in \mathcal{F},\ isOf(f, o) = true \iff \exists\ so^f \in SO.$

Definition 22. Object Frame : Let $of_o \in \mathcal{F}$ and $o \in \mathcal{O}$, of_o is an object frame for o iff $isOf(f, o)$ is true.

7.3.3 Object Frame Sequence

An object frame sequence is a finite sequence of consecutive object frames in a shot. It corresponds to the temporal trajectory followed by object in a shot. As we will see in the next chapters, this concept is very important for object-based video adaptation

applications.

Definition 23. Object Frame Sequence : Let $o \in \mathcal{O}$ an object and $sh \in \mathcal{SH}$ a shot, an object frame sequence is a finite sequence of consecutive object frames for o in sh. Formally, $ofs_o^{sh} = (f_n)_l^{n=n_0} = (f_n)_{n \in I_f^{ofs_o^{sh}}}$ is a subsequence of the shot $sh = (f_n)_{n \in I_f^{sh}}$, such that $\forall n \in I_f^{ofs_o^{sh}}$, $isOf(f_n, o)$ is true.

This definition implies that the first and last frame element of ofs_o^{sh} occurs with the appearance and disappearance of the object within a shot, respectively. Therefore, if an object appears and disappears several times in a shot, different object frame sequence are distinguished.

Definition 24. Length of an Object Frame Sequence : The length of an object frame sequence denoted by $length_{ofs_o^{sh}}$ is the number of frames constituting the object frame sequence. It is defined as $length_{ofs_o^{sh}} = |I_f^{ofs_o^{sh}}|$ such that $|I_f^{ofs_o^{sh}}|$ is the cardinality of the domain of ofs_o^{sh}.

Definition 25. Size of an Object Frame Sequence : The size of an object frame sequence denoted by $size_{ofs_o^{sh}}$ refers to the size in pixels of the object frame sequence. It is defined as the length of the object frame sequence in frames multiplied by the frame resolution, that is : $size_{ofs_o^{sh}} = length_{ofs_o^{sh}} \times size_f$.

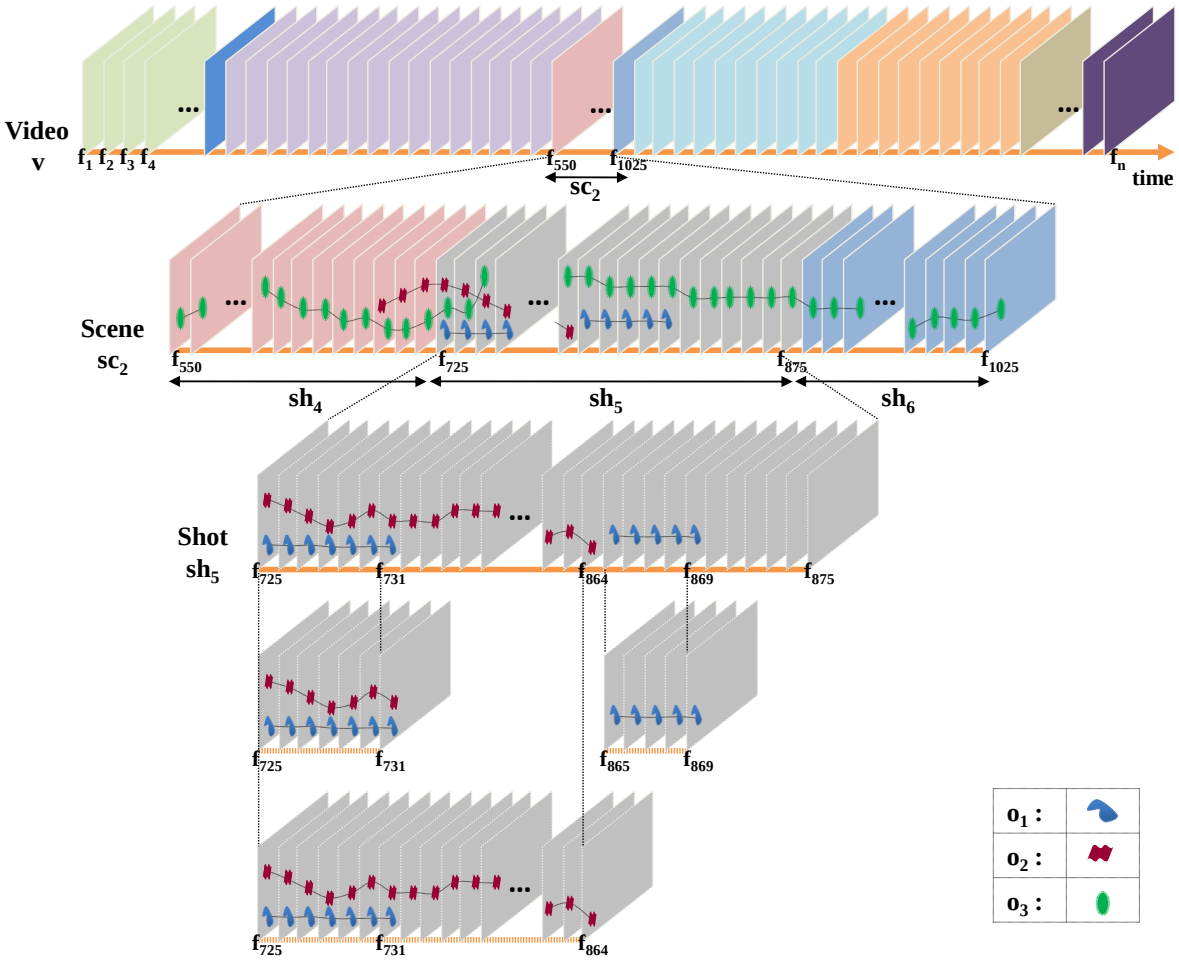


FIGURE 7.3 – Object-based video model example

Example 9. Figure 7.3 illustrates an example of the spatial and temporal struc-

ture of a video v. Based on Definition 6, the video is represented as a sequence of frames $v = (f_1,\ f_2,\ \ldots,\ f_n)$, which are ordered over the time axis. In order to differentiate between frames of different shots, we use different colors for each shot. Furthermore, each scene is represented as a sequence of shots (see Definition 17). For instance, as depicted in the figure, the scene sc_2 is defined as $sc_2 = (sh_4,\ sh_5,\ sh_6)$ such that $sh_4 = (f_{550},\ f_{551},\ \ldots,\ f_{724})$, $sh_5 = (f_{725},\ f_{726},\ \ldots,\ f_{875})$ and $sh_6 = (f_{876},\ f_{877},\ \ldots,\ f_{1025})$. Moreover, Figure 7.3 illustrates the spatial and temporal information related to two objects o_1 (i.e., with blue color) and o_2 (i.e., with red color), in the shot $sh_5 = (f_{725},\ f_{726},\ \ldots,\ f_{875})$. Two object frame sequences $ofs^{sh_5}_{o_1,1} = (f_{725},\ f_{726},\ \ldots,\ f_{731})$ and $ofs^{sh_5}_{o_1,2} = (f_{865},\ f_{866},\ \ldots,\ f_{869})$ correspond to o_1, while one frame sequence $ofs^{sh_5}_{o_2,1} = (f_{725},\ f_{726},\ \ldots,\ f_{864})$ corresponds to o_2.

7.3.4 Shot Object Frame Sequence

Definition 26. Shot Object Frame Sequence : A shot object frame sequence $SOFS^{sh}_o$ is a finite set of object frame sequences ofs^{sh}_o related to a specific object $o \in \mathcal{O}$ in a specific shot $sh \in \mathcal{SH}$. It is defined as :
$SOFS^{sh}_o = \{ofs^{sh}_o \mid ofs^{sh}_o \textit{ is an object frame sequence of } sh \textit{ related to } o\}$.

This definition implies that if an object appears in a shot sh_i and disappears in the next one sh_{i+1}, an object frame sequence for each shot is distinguished.

Going back to Example 9, $SOFS^{sh_5}_{o_1} = \{ofs^{sh_5}_{o_1,1},\ ofs^{sh_5}_{o_1,2}\}$ and $SOFS^{sh_5}_{o_2} = \{ofs^{sh_5}_{o_2,1}\}$ are the shot object frame sequence sets of sh_5 related to o_1 and o_2, respectively.

Cardinality of $SOFS^{sh}_o$: We denote by $|SOFS^{sh}_o|$ the cardinality of the finite set $SOFS^{sh}_o$.

Definition 27. Size of a Shot Object Frame Sequence : The size of a shot object frame sequence denoted by $size_{SOFS^{sh}_o}$ refers to the size in pixels of the shot object frame sequence. It is the sum of the sizes of the object frame sequences in $SOFS^{sh}_o$, that is : $size_{SOFS^{sh}_o} = \sum_{i=1}^{|SOFS^{sh}_o|} size_{ofs^{sh}_{o,i}}$.

7.3.5 Scene Object Frame Sequence

Definition 28. Scene Object Frame Sequence : A scene object frame sequence $ScOFS^{sc}_o$ related to a specific object $o \in \mathcal{O}$ in a specific scene $sc \in \mathcal{SC}$, is a finite set of $SOFS^{sh}_o$. It is defined as :
$ScOFS^{sc}_o = \{SOFS^{sh}_o \mid sh \textit{ is a shot of } sc \textit{ and } SOFS^{sh}_o \textit{ is a shot object frame sequence of } sh \textit{ related to } o\}$.

Going back to Example 9, let us assume that only two objects o_1, and o_2 appear in the shot sh_5 of the scene sc_2. Then, $ScOFS^{sc_2}_{o_1} = \{SOFS^{sh_5}_{o_1}\}$ and $ScOFS^{sc_2}_{o_2} = \{SOFS^{sh_5}_{o_2}\}$ are the scene object frame sequences in sc_2 related to o_1 and o_2, respectively.

Cardinality of $ScOFS_o^{sc}$: We denote by $|ScOFS_o^{sc}|$ the cardinality of the finite set $ScOFS_o^{sc}$.

7.3.6 Video Object Frame Sequence

Definition 29. Video Object Frame Sequence : A video object frame sequence $VOFS_o^v$ is a finite set of all scene object frame sequences $ScOFS_o^{sc}$ related to a specific object $o \in \mathcal{O}$. It is defined as $VOFS_o^v = \{ScOFS_o^{sc} \mid \forall i \in I_{sh}^{sc}, \textit{ the elements } sh_i \textit{ of } sc \textit{ are subsequences of } v\}$.

Going back to Example 9, let us assume that the two objects o_1, and o_2 only appear in the shot sh_5 of the scene sc_2. Then, $VOFS_{o_1}^v = \{ScOFS_{o_1}^{sc_2}\}$ and $VOFS_{o_2}^v = \{ScOFS_{o_2}^{sc_2}\}$ are the video object frame sequences in v related to o_1, and o_2, respectively.

Cardinality of $VOFS_o^{sh}$: We denote by $|VOFS_o^{sh}|$ the cardinality of the finite set $VOFS_o^{sh}$.

Chapter 8

Semantic Constraint Model

In this chapter, we describe the semantic constraint instantiation process responsible for filtering the video content according to the end-user semantic constraints. The aim of this process is to identify the elements of the video concerned by the end-user semantic constraints, and to provide the information (such as the position and length of the shots to be removed) needed to drive the adaptation process at a later stage. To this end, we formally define the information related to the context description process that is, the concepts of end-user semantic constraint and owner constraint. Moreover, we formally model the semantic constraint instantiation process including the information resulting from this process.

8.1 User and Owner Constraints Modelling

The context description process enables the specification of semantic constraints for the adaptation of a video content. As mentioned in Chapter 2, two types of constraints are considered : end-user semantic constraints, which let end-users specify their preferences with respect to the videos that they want to watch, and owner constraints, which let owners exercise their intellectual property rights by forbidding certain types of adaptation of their video content, and end-user semantic constraints, which let end-users specify their preferences with respect to the videos that they want to watch. For the sake of simplicity, we use the term user instead of end-user in the remainder of the thesis.

8.1.1 (User) Semantic Constraint

A (user) semantic constraint is a constraint directly related to the semantics of the video content. In the context of PIAF, we define a semantic constraint as a set of objects that the user desires to remove from the video. We propose the following definition :

Definition 30. (User) Semantic Constraint : A semantic constraint $SemCt$ is defined as a 3-tuple $SemCt = \langle name_{SemCt}, desc_{SemCt}, T_{SemCt} \rangle$, such that $name_{SemCt}$ is the name of the semantic constraint, $desc_{SemCt}$ corresponds to the description of the constraint and T_{SemCt} specifies the set of terms associated with the semantic constraints.

We refer the reader to Section 3.2.1.4 for the definitions of a term and classification scheme. In order to guaranty a consistency of the terms used to describe semantic constraints and to annotate the objects in the video, the same classification schemes are used for both. Moreover, for the purpose of this thesis, we only deal with semantic constraints related to one single object.

8.1.2 Owner Constraint

The owner preferences involve constraints related to her intellectual property rights over the content. Indeed, these constraints are expressed in terms of rules and convey explicit limitations on manipulating the video. It is worthy to note that the owner constraints are attached to a specific video. However, this is not the case for the user semantic constraints, which are defined generally in the user profile. For the purpose of this thesis, we restrict the scope to the following types of owner constraints :

1. The adaptation of the video content is forbidden ;
2. The adaptation of the video is allowed except for a specific set of shots ;
3. The adaptation of the video is allowed without removing any shots ;
4. The adaptation of the video is allowed without reducing by more than x% the original scene length.

Both type (2) and type (3) enable the owner to restrict adaptation at the shot level. We defined these types of constraints in this way, because we chose the shot as the element of the video to undergo the adaptation process. This choice is argued in detail in Section 9.1.1 of the next Chapter. On the other hand, type (4) allows the owner to impose adaptation restrictions at the scene level. It enables him to define constraints on the results of several independent adaptation decisions taken on shots composing a scene. This type of constraint is interesting for the owner, whose purpose is to protect the information transmitted by his content, because the scene represents a more semantic level. It is worthy to note that the second type of constraint can also be considered at the scene level if we forbid the adaptation of all the shots of a scene.

We can already draw some conclusion on the way to process these types of constraints by analysing them. For type (1), it is clear that the video cannot be adapted and no further processing is required. For type (2), if the user semantic constraint requires an adaptation of shots among those referenced by the owner constraint, then the adaptation is not possible. Otherwise, the adaptation process can be performed without restrictions. Type (3) forbids the removal of all frames contained in any shot, but allows their modification. Therefore, it restricts the type of adaptation operation that can be considered in the adaptation process.

8.2 Formal Modelling of the Semantic Constraint Instantiation Process

In PIAF, when the user requests a video, a semantic constraint instantiation process takes place with the goal of filtering the video content according to the user semantic constraints. We note that this process is done independently from the owner constraint. These latter are taken into consideration afterwards during the adaptation process. To begin with, the instantiation process queries the user semantic constraints to extract its list of terms. Then, it parses the content description of the requested video in order to identify the objects with which the terms match. Once the objects are identified, the video description is filtered to determine the spatial and temporal information related to the identified objects. At this stage, the filtered descriptions do not contain all the information required to drive the adaptation process. Indeed, the length of a shot, scene and object frame sequences, and the size of an object in a frame must be calculated based on the information provided in the filtered content description.

The output of this process, so-called *InstantiatedConstraint*, consists of all the information needed to drive the adaptation process at a later stage. In the remainder of this section, we formally model the semantic constraint instantiation process including the information resulting from it. For the sake of simplicity, we only deal with one semantic constraint at a time. In case of several semantic constraints, the same process is applied for each constraint separately.

8.2.1 Set of Scenes to be Adapted within a Video

Given a video v, a semantic constraint $SemCt$ and the object $o \in \mathcal{O}$ identified by the instantiation process, the set of the scenes within the video v that contain o and must undergo the adaptation process is given by the function $Adp\mathcal{SC}_v$ defined as follows :

$$\begin{array}{rcccl} Adp\mathcal{SC}_v & : & \mathcal{O} & \longrightarrow & \mathcal{P}(\mathcal{SC}) \\ & & o & \longmapsto & Adp\mathcal{SC}_v(o) = \{\ sc \mid sc \in \mathcal{SC}\ and\ ScOFS_o^{sc} \neq \emptyset\} \end{array} \tag{8.1}$$

whereas the codomain of the function $Adp\mathcal{SC}_v$ is the powerset of the set $\mathcal{SC}$. We refer the reader to Section 7.3.5 for the definition of $ScOFS_o^{sc}$. In the rest of the thesis, we use the simplified notation $Adp\mathcal{SC}_{v,o}$ instead of $Adp\mathcal{SC}_v(o)$.

Cardinality of $Adp\mathcal{SC}_{v,o}$: We denote by $|Adp\mathcal{SC}_{v,o}|$ the cardinality of the finite set $Adp\mathcal{SC}_{v,o}$.

Example 10. Going back to Example 9 illustrated in Figure 7.3, let v be a video with the two sets $\mathcal{SH} = \{sh_1,\ sh_2,\ \ldots,\ sh_{10}\}$ and $\mathcal{SC} = \{sc_1 = (sh_1,\ sh_2,\ sh_3),\ sc_2 = (sh_4,\ sh_5,\ sh_6),\ \ldots,\ sc_4 = (sh_{10})\}$. Let us assume that $VOFS_{o_2}^{v} = \{ScOFS_{o_2}^{sc_2}\}$ such that $ScOFS_{o_2}^{sc_2} = \{SOFS_{o_2}^{sh_4},\ SOFS_{o_2}^{sh_5}\}$ for $o_2 \in \mathcal{O}$. Thus, $Adp\mathcal{SC}_{v,o_2} = \{sc_2\}$ with cardinality $|Adp\mathcal{SC}_{v,o_2}| = 1$.

Given the set of the scenes $Adp\mathcal{SC}_{v,o}$ to be adapted, for each $sc \in Adp\mathcal{SC}_{v,o}$, we can compute its length in number of frames $length_{sc}$ (see Definition 18). Accordingly, we

can derive the set of shots to be adapted in each scene as explained in the next section.

8.2.2 Set of Shots to be Adapted within a Scene

Given a scene $sc \in Adp\mathcal{SC}_{v,o}$ and the related object $o \in \mathcal{O}$ to be removed, the set of shots within the scene sc that contain o and so must undergo the adaptation process is given by the function $Adp\mathcal{SH}$ defined as follows :

$$\begin{array}{rccl} Adp\mathcal{SH} & : & \mathcal{SC} \times \mathcal{O} & \longrightarrow & \mathcal{P}(\mathcal{SH}) \\ & & (sc, o) & \longmapsto & Adp\mathcal{SH}(sc, o) = \{\ sh \mid sh \in \mathcal{SH} \text{ and } SOFS_o^{sh} \neq \emptyset \text{ and} \\ & & & & sh \text{ is an element of } sc\} \end{array} \tag{8.2}$$

whereas the codomain of the function $Adp\mathcal{SH}$ is the powerset of the set $\mathcal{SH}$. We refer the reader to Section 7.3.4 for the definition of $SOFS_o^{sh}$.
In the rest of the thesis, we use the simplified notation $Adp\mathcal{SH}_{sc,o}$ instead of $Adp\mathcal{SH}(sc, o)$.

Cardinality of $Adp\mathcal{SH}_{sc,o}$: We denote by $|Adp\mathcal{SH}_{sc,o}|$ the cardinality of the finite set $Adp\mathcal{SH}_{sc,o}$.

Example 11. Going back to Example 10, given $o_2 \in \mathcal{O}$ and $sc_2 \in \mathcal{SC}$, we define $Adp\mathcal{SH}_{sc_2,o_2} = \{sh_4,\ sh_5\}$ with cardinality $|Adp\mathcal{SH}_{sc_2,o_2}| = 2$.

Given the set of the shots $Adp\mathcal{SH}_{sc,o}$ to be adapted for a scene $sc \in \mathcal{SC}$, for each $sh \in Adp\mathcal{SH}_{sc,o}$, we can compute its length in number of frames $length_{sh}$ (see Definition 13) and its size in number of pixels $size_{sh}$ (see Definition 14).
Furthermore, we define a mapping function between the set of indices of the shots $sh \in Adp\mathcal{SH}_{sc,o}$ and the set $\{1,\ 2,\ \ldots,\ |Adp\mathcal{SH}_{sc,o}|\}$. Indeed, this function is required to enable an iteration over the elements sh in the set $Adp\mathcal{SH}_{sc,o}$ in ascending order (see Section 9.2.6 and Section 10.1).

Set of indices for the $Adp\mathcal{SH}_{sc,o}$ elements : given a scene $sc \in \mathcal{SC}$ and an object $o \in \mathcal{O}$ to be removed from a video v, we define the function Idx given by :

$$\begin{array}{rccl} Idx & : & \mathcal{SC} \times \mathcal{O} & \longrightarrow & \mathcal{P}(I_{sh}^v) \\ & & (sc, o) & \longmapsto & Idx(sc, o) = \{i \in I_{sh}^v \mid sh_i \in Adp\mathcal{SH}_{sc,o}\} \end{array} \tag{8.3}$$

where the codomain is the powerset of the set of the shot indices I_{sh}^v, and $Idx(sc, o)$ is the set of indices for the $Adp\mathcal{SH}_{sc,o}$ elements.

Index correspondence function τ : For each $(sc, o) \in \mathcal{SC} \times \mathcal{O}$, we denote by $\tau_{sc,o}$ the bijective function from the set $Idx(sc, o)$ to the set $\{1,\ 2,\ \ldots,\ |Adp\mathcal{SH}_{sc,o}|\}$. We denote by $\tau_{sc,o}^{-1}$ the inverse function of $\tau_{sc,o}$. The functions are defined as follows :

$$Idx(sc, o) \underset{\tau_{sc,o}^{-1}}{\overset{\tau_{sc,o}}{\rightleftarrows}} \{1,\ \ldots,\ |Adp\mathcal{SH}_{sc,o}|\} \tag{8.4}$$

Example 12. Going back to example 11, we define the set $Idx(sc_2, o_2) = \{4,\ 5\}$ for $(sc_2, o_2) \in \mathcal{SC} \times \mathcal{O}$. Thus, the bijective function τ_{sc_2,o_2} is given by :

$$\{4,\ 5\} \ \underset{\tau^{-1}_{sc_2,o_2}}{\overset{\tau_{sc_2,o_2}}{\rightleftarrows}} \ \{1,\ 2\}$$

such that $\tau_{sc_2,o_2}(4) = 1$, $\tau_{sc_2,o_2}(5) = 2$, $\tau^{-1}_{sc_2,o_2}(1) = 4$ and $\tau^{-1}_{sc_2,o_2}(2) = 5$.

Although we are aware of the need of the $\tau^{-1}_{sc,o}$ function, we assume for clarity of the presentation that $\tau_{sc,o} = \tau^{-1}_{sc,o} = identity$, in the remainder of the thesis.

8.2.3 Spatial and Temporal Information related to the Object

Once the semantic instantiation process has identified the set of shots to be adapted, the temporal and spatial information related to the object must be extracted or computed, for each considered shot.

The temporal information consists of the length and size of both object frame sequences and shot object frame sequences. We refer the reader to Section 7.3.3 and 7.3.4 for their definitions and their properties.

The spatial information consists of the size of the spatial object in each object frame of the shot. We refer the reader to Section 7.3.1 for the definition of spatial region and its properties.

Chapter 9

Metadata-driven Utility-based Adaptation Engine

In this chapter, we describe the process applied by our PIAF Adaptation Decision-Taking Engine to compute an appropriate adaptation plan for a given instantiated constraint. We first introduce notations and provide background on the fundamental concepts associated with the adaptation (i.e., entity, adaptation operations, constraints and utility). Then, we discuss the relationships among these concepts and describe the semantic object-based adaptation procedure. Afterwards, we present our Utility Function (UF) model that computes a utility value for each entity affected by the constraint and for each adaptation operation. This UF models the relationships between five quality parameters in order to provide a global quality value. We present the five quality parameters and conclude this chapter with an output format of the UF.

9.1 Semantic Object-based Video Adaptation

9.1.1 Definition of the Adaptation Concepts

In this section, we define the fundamental concepts associated with the semantic object-based video adaptation, which are used in the remainder of this chapter.

The set of granularity levels in a video : adaptation operations on a video can be performed at different granularity levels. A granularity level corresponds to a level in the video model presented in Chapter 7. The set of the granularity levels denoted $\mathcal{G}$ therefore comprises pixels, region, spatial object, frame, object frame sequences, object frame sequences, shot, scenes, etc.

Definition 31. Entity : As defined in [20], an entity is the basic unit of a video element that undergoes the adaptation process, and it may exist at different granularity levels.

For the purpose of this work, we define the entity based on several requirements. Since we aim to make a spatio-temporal adaptation reasoning, the entity should be a sequence of frames sharing the same semantic properties. This requirement restricts the choice to shots and scenes. Moreover, since one single adaptation operation could be applied over an entity, the size of the entity and the amount of information enclosed

in it are criteria that could affect the performance of the adaptation decision. For instance, let us consider a scene of two shots to be adapted. Assume that the size of the object to be removed is too large in the first shot and small in the second shot. A logical reasoning would be to simply remove the first shot, but remove the object from the frames of the second shot. Taking into consideration the requirement 'one single adaptation operation could be applied over an entity', the result of adapting the scene will be unpleasant if we consider the scene as an entity. For instance, if we apply a removal of the object, this will yield artifacts in the first shot. Moreover, if we apply a removal of the shots, this will yield a loss of information as the whole scene will be removed. To this end, we define the shot as the entity undergoing the adaptation process.

The set of all adaptation methods : We denote by $\mathcal{A}$ the set of all adaptation methods. Some examples of adaptation methods are : scaling, selection, reduction, removal, merging, replacement, synthesis, etc. We refer the reader to [20] and [71] for a comprehensive discussion of adaptation methods.

In this thesis, we focus on the removal adaptation methods denoted by *Drop*. Thus, the set $\mathcal{A}'$ of the considered adaptation methods is a singleton, $\mathcal{A}' = \{Drop\} \subset \mathcal{A}$

Definition 32. Adaptation Operator : An adaptation operator *op* is defined as a pair of $(a_i, g_j) \in \mathcal{A} \times \mathcal{G}$. We note $\mathcal{OP} = \{(a_i, g_j) \ / \ 1 \leq i \leq \mid \mathcal{A} \mid \ and \ 1 \leq j \leq \mid \mathcal{G} \mid\}$ the set of all adaptation operators defined over $\mathcal{A} \times \mathcal{G}$.

Since the shot is defined as an entity, we thus restrict ourselves to the granularity set $\mathcal{G}' = \{spatial\ object,\ shot\ object\ frame\ sequence,\ shot\}$.

Consequently, we considered the restricted set of adaptation operators $\mathcal{OP}'$:
$\mathcal{OP}' = \{(Drop, g'_j) \ / \ 1 \leq j \leq \mid \mathcal{G}' \mid\}$
$= \{(Drop, Spatial\ Object),\ (Drop, Shot\ Object\ Frame\ Sequence),\ (Drop, Shot)\}$
such that $|\mathcal{G}|$ is the cardinality of set $\mathcal{G}'$. In the rest of the thesis, we use the simplified notation for the elements of $\mathcal{OP}'$, that is : $\mathcal{OP}' = \{(Drop - Object),\ (Drop - SOFS),\ (Drop - Shot)\}$

Definition 33. Adaptation Operation : An adaptation operation transforms an entity into a new one by applying an adaptation operator to the entity.

Given an object o to be removed from a shot sh to be adapted and a operator $op \in \mathcal{OP}'$, the effect of applying an *op* to sh is discussed in the following :

- ***op = (Drop-Object)*** means : for all the frames existing in the shot sh, each frame should not contain the set of spatial objects related to a specific object o defined by a term t, and if the frame is an object frame, the spatial object should be removed from the frame by an object removal operation.
- ***op = (Drop-SOFS)*** means : all the concerned Object Frame Sequences related to this object and existing in the shot should be removed.
- ***op = (Drop-Shot)*** means : the shot should be removed, or equivalently, all the frames contained in this shot should be removed.

Definition 34. Utility : Relying on the definition in MPEG-21 DIA [16], a utility denoted u is a measurement of the QoE resulting from an adaptation operator over an entity.

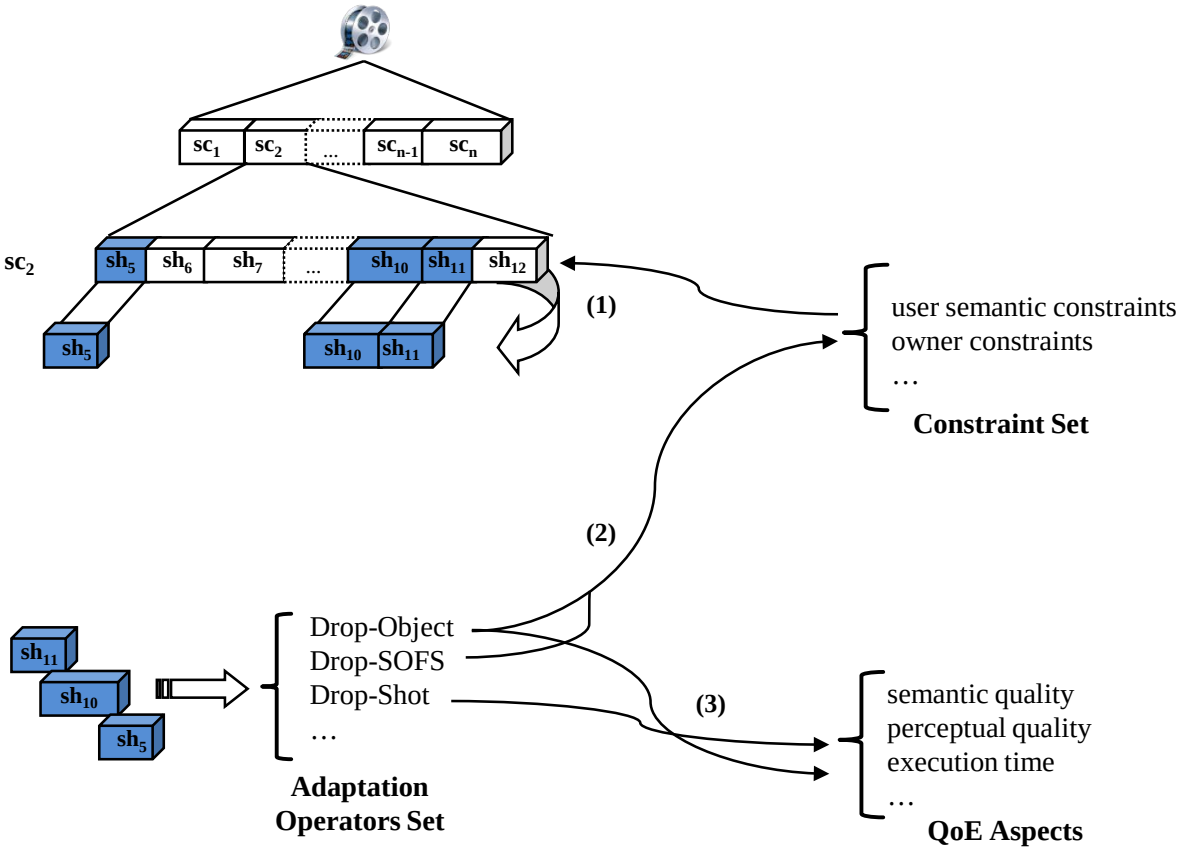


FIGURE 9.1 – Semantic object-based video adaptation procedure.

9.1.2 Semantic Object-based Video Adaptation Procedure

Figure 9.1 illustrates the procedure of the semantic adaptation performed in PIAF. This procedure is based on the general conceptual framework for video adaptation described in [20]. In the following, we describe the steps for finding the best combination of adaptation operators within a scene in order to satisfy the semantic constraints specified by the user while respecting the owner's constraints. In order to compute the best adaptation plan for a video, this process should be iterated over all the scenes of a video.

(1) Based on the information described in the instantiation constraint, for each scene, identify the shots that must be adapted to satisfy the owner's and user's constraints set.

(2) For each identified shot, select the adaptation operators that do not conflict with the owner constraints. In order to satisfy the user's constraints, several adaptation operators can be applied to the shot. However, the owner can restrict the use of some operators. For instance, she can allow the adaptation of the video, but without removing any shots. Thus, the adaptation operator 'Drop-Shot' is not a feasible one.

(3) As UME is targeted, the selection of one of these adaptation operators for each shot must be guided by the goal of maximizing the QoE that the user will gain from the adapted content. As discussed in Section 2.4.2, the QoE is calculated by a quality metric. Depending on the application scenario-based adaptation, the quality metric can be modelled to consider one or more of the QoE aspects such as semantic quality (refer to Section 3.1.5.1), perceptual quality (refer to

Section 3.1.5.2), execution time of the adaptation operation, etc. For each adaptation operation, this metric results in a utility value that measures the QoE resulting from the adaptation operator over the shot. Thus, to select the best adaptation operator, we need to measure the utility value associated with video shots undergoing the identified adaptation operators and then choose the one with the highest value.

Accordingly, to enable a semantic object-based video adaptation based on this procedure, two major research questions are to be addressed :

(1) How to model a quality metric that can predict a global quality of experience value, which is consistent with subjective evaluation ? What are the quality aspects to be considered and how can each of them be computed along different adaptation operators ?

(2) How to compute the best adaptation plan that satisfies both the user's and the owner's constraints ?

In the remainder of this chapter, we give our solutions to the first question. To begin with, we develop a utility function that computes a utility value from five different quality parameters [31] [32] : affected area, affected priority area, temporal perceived quality, spatial perceived quality and processing cost. Then, we define functions to compute each of the five quality parameters along each of the adaptation operations described in Definition 33. We also show how these quality parameters capture three aspects of the QoE : semantic quality, perceptual quality and execution time. Finally, we present the output of the utility value computation process. More details about the utility function and the parameters are given in the next Section 9.2.

9.2 Utility Function

We define the Utility Function (UF) as a linear function, whose parameters are non-uniformly weighted. As previously stated, the goal is to capture the different parameters that may affect the quality of experience provided by the adapted content. Since we are considering semantic constraints that require object removal, we have to cope with a loss of quality from the original to the adapted content. Thus, the purpose of the different parameters is to represent and evaluate different causes of quality loss.

First, we assume that the less the content is modified by the adaptation, the better it is. Thus, we define a parameter p_1, which evaluates the surface affected by an adaptation. Nevertheless, the adaptation of two entities of the same area size can have a different impact on the comprehension of the content. Indeed, some parts of the video could be more important than others with respect to the semantic content. Thus, we define a parameter p_2 to model the priority of the affected area ; p_2 makes use of the notion of priority of shots and objects (see Section 9.2.3.1). We assume that adapting the content while preserving the highest priority items should better preserve the semantic integrity of the original video. Furthermore, as users may be disturbed by 'gaps' created in the temporal domain when deleting shots or Shot Object Frame Sequence (SOFS), we define a parameter p_3 to evaluate the loss in temporal quality. Note that the *Drop-Object* operation is not concerned with this parameter, as it does not create gaps. On the other hand, its implementation, using removal techniques, may cause losses of visual quality in the adapted content. Thus, we define a parameter p_4

to anticipate this effect. Finally, as users may prefer to trade off a lower quality for a smaller waiting time, we introduce a parameter p_5 to measure the processing cost of each operation. We denote by $\mathcal{P}$ the set of parameters given by :

$$\mathcal{P} = \left\{ \begin{array}{c} p_1 : affected\ area,\ p_2 : affected\ priority\ area,\ p_3 : temporal\ perceived \\ quality,\ p_4 : spatial\ perceived\ quality, p_5 : processing\ cost \end{array} \right\}$$

where each parameter $p_i \in \mathcal{P}$ is defined as $p_i : \mathcal{OP}' \times \mathcal{SH} \rightarrow [0,1]$ such that the higher the value returned by the function, the higher is the utility value of the operation with respect to the considered parameter.

Furthermore, the UF evaluates the impact of possible adaptation operations on a shot along several parameters and combines their values in a single one. This combination is performed based on assigning a weight to each parameter, thereby making it possible to tune the contribution of each parameter in the utility for different cases and contexts. The resulting utility value allows for a quantitative comparison to be made between different adaptation operators.

Definition 35. Utility Function : We define the utility function as $UF : \mathcal{OP}' \times \mathcal{SH} \rightarrow [0,1]$ such that $\forall op \in \mathcal{OP}'$, $\forall sh \in \mathcal{SH}$, $UF(op, sh)$ is the weighted means of the data set $\{p_1(op, sh), \ldots, p_k(op, sh)\}$ with the non-negative weights $\{w_1, \ldots, w_k\}$:

$$UF(op, sh) = \frac{\sum_{i=1}^{k}(w_i \times p_i(op, sh))}{\sum_{i=1}^{k} w_i} \tag{9.1}$$

k denotes the number of parameters p_k used in the utility function (actually 5 parameters), and each parameter is normalized to vary between 0 and 1. The weights $w_1, \ldots, w_k$, respectively, refer to the influence of the parameter $p_1(op, sh), \ldots, p_k(op, sh)$ on the UF. Their values were experimentally fixed independently of the shots, such that $\sum_{i=1}^{k} w_i = 1$ with $k = 5$. For instance, Section 13.4.2.1 describes the process that finds the values of the weights. Accordingly, the utility function is expressed as :

$$UF(op, sh) = \sum_{i=1}^{5}(w_i \times p_i(op, sh)) \tag{9.2}$$

9.2.1 Affected Area Ratio

We introduce the notion of affected area ratio, which is used in several parts of the utility function. The affected area ratio function denoted by *aar* is defined as the ratio of the exact size of the area affected by an adaptation operation in pixels divided by the exact size of the area of the shot in pixels :

$$\begin{array}{rccl} aar : & \mathcal{OP}' \times \mathcal{SH} & \longrightarrow & [0,1] \\ & (op, sh) & \longrightarrow & aar(op, sh) = \dfrac{sizeof(affected\ area)}{sizeof(sh)} \end{array} \tag{9.3}$$

It is worthy to note that the area affected by an adaptation operation, noted $affected\ area$, is related to the granularity $g_j \in \mathcal{G}$ of the pair $(Drop, g_j) \in \mathcal{OP}'$. According to this definition, the value of $aar(Drop - Shot, sh)$ is evidently equal to 1. For $op = Drop - SOFS$, we calculate the size of each affected $ofs_o^{sh} \in SOFS_o^{sh}$ (refer to Section 7.3.3 and Section 7.3.4). Thus, $aar(Drop - SOFS, sh)$ is given by :

$$\begin{aligned} aar(Drop - SOFS, sh) &= \frac{size_{shofs}}{size_{sh}} = \frac{\sum_{i=1}^{|SOFS_o^{sh}|} size_{ofs_{o,i}^{sh}}}{size_{sh}} \\ &= \frac{\sum_{i=1}^{|SOFS_o^{sh}|} length_{ofs_{o,i}^{sh}} \times size_f}{length_{sh} \times size_f} = \frac{\sum_{i=1}^{|SOFS_o^{sh}|} length_{ofs_{o,i}^{sh}}}{length_{sh}} \end{aligned} \tag{9.4}$$

Moreover, for $op = Drop - Object$, we calculate the size of the spatial object so in each frame of $ofs_o^{sh} \in SOFS_o^{sh}$. Thus, $aar(Drop - Object, sh)$ is given by :

$$aar(Drop - Object, sh) = \frac{\sum_{i=1}^{|SOFS_o^{sh}|} \sum_{j \in I_f^{ofs_{o,i}^{sh}}} size_{so^{f_j}}}{length_{sh} \times size_f} \tag{9.5}$$

where $size_{so^{f_j}}$ is the size of the spatial region in the frame f_j of the i^{th} $ofs_{o,i}^{sh} \in SOFS_o^{sh}$ within a shot sh.

To summarize, given an object $o \in \mathcal{O}$, $\forall sh \in \mathcal{SH}$, $and\ \forall op \in \mathcal{OP}'$, we define $aar(op, sh)$ as follows :

$$aar(op, sh) = \begin{cases} \dfrac{size_{sh}}{size_{sh}} = 1 & , \ if\ op = (Drop - Shot) \\ \dfrac{\sum_{i=1}^{|SOFS_o^{sh}|} length_{ofs_{o,i}^{sh}}}{length_{sh}} & , \ if\ op = (Drop - SOFS) \\ \dfrac{\sum_{i=1}^{|SOFS_o^{sh}|} \sum_{j \in I_f^{ofs_{o,i}^{sh}}} size_{so^{f_j}}}{length_{sh} \times size_f} & , \ if\ op = (Drop - Object) \end{cases} \tag{9.6}$$

In the remainder of this thesis, we use the aar function without detailing it for sake of simplification.

9.2.2 Affected Area

Parameter p_1 aims to evaluate the surface of the area affected by an adaptation operation. Thus, it calculates the number of all pixels affected by the adaptation operation relative to the number of all pixels in the shot. We define p_1 as the function

$p_1 : \mathcal{OP}' \times \mathcal{SH} \rightarrow [0,1]$ such that $\forall sh \in \mathcal{SH},\ p_1(op, sh)$ is given as follows :

$$\begin{aligned} p_1(op, sh) &= 1 - \frac{number\ of\ all\ affected\ pixels\ in\ the\ shot}{number\ of\ all\ pixels\ in\ the\ shot} \\ &= 1 - aar(op, sh) \end{aligned} \tag{9.7}$$

The subtraction from 1 is due to the fact that the higher the value returned by p_1, the higher is the utility value of the operation with respect to p_1 (refer to Section 9.2) will be. Accordingly, given an object $o \in \mathcal{O},\ \forall sh \in \mathcal{SH},\ and\ \forall op \in \mathcal{OP}'$, we define $p_1(op, sh)$ as follows :

$$p_1(op, sh) = \begin{cases} 1 - \dfrac{size_{sh}}{size_{sh}} = 0 & ,\ if\ op = (Drop - Shot) \\ 1 - aar(Drop - SOFS, sh) & ,\ if\ op = (Drop - SOFS) \\ 1 - aar(Drop - Object, sh) & ,\ if\ op = (Drop - Object) \end{cases} \tag{9.8}$$

9.2.3 Affected Priority Area

Parameter p_2 aims to calculate the affected priority area ratio for each adaptation operation, based on the priority value assigned to the object and to the shot during the annotation process. This parameter measures the ratio of the exact size of an affected priority zone in pixels over the exact size of the priority zone in pixels. Therefore, we define p_2 as $p_2 : \mathcal{OP}' \times \mathcal{SH} \rightarrow [0,1]$ such that :

$$p_2(op, sh) = 1 - \frac{size\ of\ the\ affected\ area\ of\ the\ priority\ zone\ in\ the\ shot}{size\ of\ the\ priority\ zone\ in\ the\ shot} \tag{9.9}$$

Before going further into the explanation of the affected priority area, it is worth explaining how the priority value is assigned to the ofs_o^{sh} and $SOFS_o^{sh}$, given the priority of both the object and the shot. Moreover, we explain in detail how the priority zone and the affected priority zone are computed in a shot so as to find the value of the affected priority area ratio.

9.2.3.1 Priority

At the shot and object level, a semantic priority value is assigned during the annotation process (refer to Chapter 11). For the purpose of this thesis, we consider a binary priority value of 0 or 1. We define functions $\rho : \mathcal{SH} \rightarrow \{0,1\}$ and $\rho' : \mathcal{O} \rightarrow \{0,1\}$ such that $\rho(sh)$ and $\rho'(o)$ are 1 if the shot or object is of high priority, and 0 otherwise. For the sake of simplicity, we denote by ρ_{sh} and ρ'_o the priority of the shot sh and object o, respectively.

It is important to note that there is no absolute dependency between the priority of a shot and the priority of the objects that appear in it. For instance, Figure 9.2 illustrates an excerpt of images of the first five shots from the video used in Section 2.4.1. The

woman is a prior object in the story ; however, some shots in which she appears have a priority value of 0 (i.e., sh_3 and sh_4). Furthermore, although the apple logo is not a prior object in the story, the shot in which it appears is prior (i.e., sh_5).

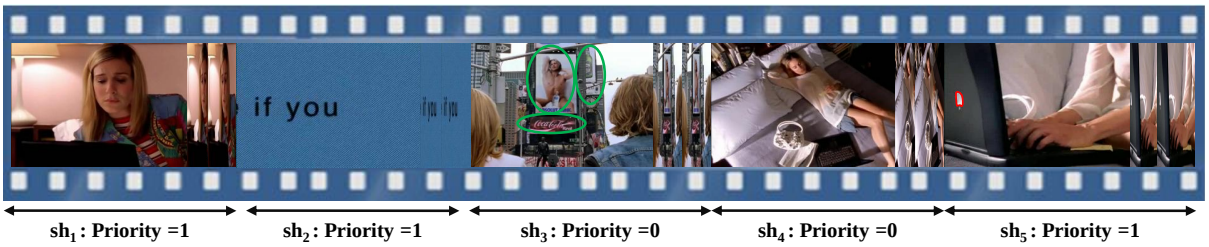


FIGURE 9.2 – Illustration of the priority information in a video.

Accordingly, we give the following priority definitions :

Definition 36. : Given $\mathcal{O}_{sh} \subset \mathcal{O}$, the set of objects appearing in the shot sh, having the priority of the shot does not imply that we can infer the priority of the object, and vice-versa.

$$Given\ \rho'_o\ and\ \rho_{sh},\ \forall\ pval \in \{0,1\}\quad \rho_{sh} = pval \not\Leftrightarrow\ \rho'_o = pval$$

Definition 37. Shot Object Frame Sequence Priority : The priority value of an $SOFS_o^{sh}$ is inferred from the priority value of both the object o and the shot sh. We define the priority of an $SOFS_o^{sh}$ as the function $\phi : \mathcal{SOFS} \rightarrow \{0,1\}$ such that $\forall\ sh \in \mathcal{SH}$ and $o \in \mathcal{O}$ appearing in sh, and thus we have :

$$\phi(SOFS_o^{sh}) = \begin{cases} \rho_{sh} \quad , & if\ \rho_{sh} = 1 \\ \\ \rho'_o \quad , & if\ \rho_{sh} = 0 \end{cases}$$

such that $\mathcal{SOFS}$ is the set of all the shot object frame sequence in a video (see Section 7.3.4). In the rest of the thesis, we use the simplified notation $\phi_{SOFS_o^{sh}}$ instead of $\phi(SOFS_o^{sh})$.

Definition 38. Object Frame Sequence Priority : Given the priority value of a shot object frame sequences, all its elements inherit the same priority value. Let $\phi'_{ofs_o^{sh}}$ denote the priority value of an object frame sequence $ofs_o^{sh} \in size_{ofs_o^{sh}}$, where $\phi'_{ofs_o^{sh}}$ is defined as follows :

$$Given\ \phi_{size_{ofs_o^{sh}}},\ \forall\ ofs_o^{sh} \in SOFS_o^{sh},\ \phi'_{ofs_o^{sh}} = \phi_{SOFS_o^{sh}}$$

Example 13. Let $\mathcal{O}_{sh_1} = \{o_1,\ o_2,\ o_3\} \subset \mathcal{O}$ be the set of objects appearing in the shot $sh_1 \in \mathcal{SH}$ such that $\rho' o_1 = \rho' o_3 = 1$ and $\rho' o_2 = 0$. As depicted in Figure 9.3, the sets $SOFS_{o_1}^{sh_1} = \{ofs_{o_1,1}^{sh_1},\ ofs_{o_1,2}^{sh_1}\}$, $SOFS_{o_2}^{sh_1} = \{ofs_{o_2,1}^{sh_1}\}$ and $SOFS_{o_3}^{sh_1} = \{ofs_{o_3,1}^{sh_1}\}$ are the shot object frame sequence sets related to o_1, o_2 and o_3, respectively.

According to the previous properties and definitions, if $\rho_{sh_1} = 1$ then $\phi_{SOFS_{o_1}^{sh_1}} = \phi_{SOFS_{o_2}^{sh_1}} = \phi_{SOFS_{o_3}^{sh_1}} = \rho_{sh_1} = 1$. However, if $\rho_{sh_1} = 0$ then $\phi_{SOFS_{o_1}^{sh_1}} = \rho' o_1 = 1$, $\phi_{SOFS_{o_2}^{sh_1}} = \rho' o_2 = 0$ and $\phi_{SOFS_{o_3}^{sh_1}} = \rho' o_3 = 1$. By inference the priority value of all the $SOFS$s elements $\phi'_{ofs_{o_1,1}^{sh_1}}$, $\phi'_{ofs_{o_1,2}^{sh_1}}$, $\phi'_{ofs_{o_2,1}^{sh_1}}$ and $\phi'_{ofs_{o_3,1}^{sh_1}}$, inherit the value of their shot object frame sequence. For instance, for $\rho_{sh_1} = 1$, their value will be equal to 1.

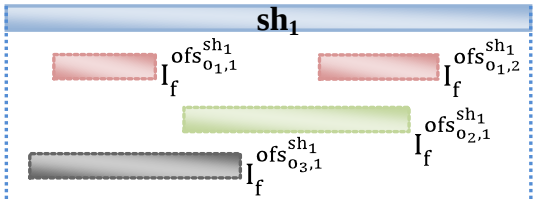


FIGURE 9.3 – Object frame sequence sets related to three objects o_1, o_2 and o_3 in a sh_1.

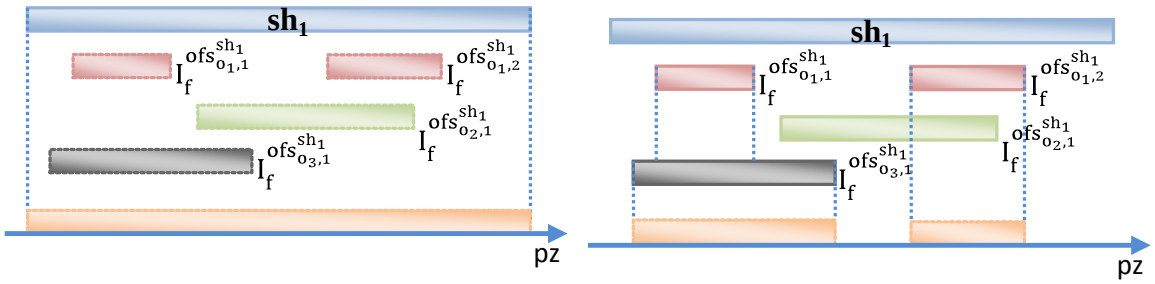
9.2.3.2 Priority Zone

The priority zone within a shot is the union of the domains of the object frame sequences (i.e., video intervals as defined in Property 4), which are related to all the objects appearing in the shot, and have a priority value equal to 1. Figure 9.4 illustrates the calculation of the priority zone. If $\rho_{sh} = 1$ then we define the priority zone to correspond to the domain of the shot. Otherwise if $\rho_{sh} = 0$, then the priority zone is computed by Algorithm 9.1 that is described as follows :
Given a shot $sh \in \mathcal{SH}$, let $Prior = \{I_f^{ofs_o^{sh}} \mid ofs_o^{sh}$ *is a shot object frame sequence of* sh *related to* o *and* $\phi'_{ofs_o^{sh}} = 1\}$ be the set of the domains of the object frame sequences related to all the objects appearing in the shot, and have a priority value equal to 1. The algorithm takes $Prior$ as an input and finds the union of the intervals using the 13 temporal relations of Allen [6]. The result of the algorithm is a set of non-overlapping intervals D_i, which is the set of domains of the prior frame sequences fss that we are searching for.

Based on this definition, the priority zone pz is given by the function :
$pz : \mathcal{SH} \rightarrow \mathcal{P}(\mathcal{D}(I_f^v))$ whose codomain is the powerset of the set $\mathcal{D}(I_f^v)$(see Definition 8) :

$$pz(sh) = \begin{cases} I_f^{sh} & , \quad if\ \rho_{sh} = 1 \\ \bigcup_{i=1}^{k} D_i & , \quad if\ \rho_{sh} = 0 \end{cases} \tag{9.10}$$

where k is the cardinality of the set of intervals D_i resulting from Algorithm 9.1.



(a) $\rho_{sh_1} = 1$ (b) $\rho_{sh_1} = 0$

FIGURE 9.4 – Calculating the priority zone in a shot.

Accordingly, we define the size of the priority zone in pixels denoted by $\delta(pz(sh))$

as follows :

$$\delta(pz(sh)) = \begin{cases} length_{sh} \times size_f & , \ if \ \rho_{sh} = 1 \\ \sum_{i=1}^{k} length_{D_i} \times size_f & , \ if \ \rho_{sh} = 0 \end{cases} \quad (9.11)$$

where $length_{sh}$ is the length in number of frames of the shot and $length_{D_i}$ is the length in number of frames of the intervals D_i resulting from the algorithm, whereas $i = 1,\ 2,\ \ldots,\ k$ and k is the cardinality of the resulting set.

Example 14. Back to Example 13, let us suppose that $sh_1 = (f_1,\ f_2,\ \ldots,\ f_{120})$, such that $\rho_{sh_1} = 0$, $ofs^{sh_1}_{o_1,1} = (f_{10},\ f_{11},\ \ldots,\ f_{35})$, $ofs^{sh_1}_{o_1,2} = (f_{74},\ f_{75},\ \ldots,\ f_{100})$, $ofs^{sh_1}_{o_2,1} = (f_{41},\ f_{42},\ \ldots,\ f_{95})$ and $ofs^{sh_1}_{o_3,1} = (f_5,\ f_6,\ \ldots,\ f_{55})$. In the following, we describe the steps of Algorithm 9.1 whose input is the set $Prior = \{[10, 35],\ [74, 100],\ [5, 55]\}$. The result of the algorithm is illustrated in Figure 9.4b.

Input :
- $PriorList = \langle 10, 35\rangle : \langle 74, 100\rangle : \langle 5, 55\rangle$

Initialize variables :
- init($PZList$)

Step 1 :
- Sort the elements of $PriorList$ in an ascending order of their lower endpoint : $PriorList = \langle 5, 55\rangle : \langle 10, 35\rangle : \langle 74, 100\rangle$
- Initialize $q = \langle 5, 55\rangle$
- Update $PriorList = \langle 10, 35\rangle : \langle 74, 100\rangle$

Step 2 : While $PriorList$ is not empty, add the elements of $PriorList$ according to the 13 temporal relations of Allen.

Execute the 1^{st} iteration :
- Set $p = \langle 10, 35\rangle$. It corresponds to Allen's temporal relation **p during q**. Thus, p is ignored and removed from the $PriorList$.
- Update $PriorList = \langle 74, 100\rangle$

Execute the 2^{nd} iteration :
- Set $p = \langle 74, 100\rangle$. It corresponds to Allen's temporal relation **p takes place after q**. Thus, p is added to $PZList$.
- Update $PZList = \langle 74, 100\rangle$
- Update $PriorList$ is empty

Step 3 :
- Update $PZList = \langle 5, 55\rangle : \langle 74, 100\rangle$
- Return $PZList$

Output :
- $PZList = \langle 5, 55\rangle : \langle 74, 100\rangle$

Definition of the Variables, Functions and Data types of the Algorithm 9.1.

Variables :

$PZList$, $PriorList$ **:** list of type $ListIntervals$.

p, q **:** pointers to element of type $Interval$.

Data Types :

$Interval$ {
fs : the lower endpoint of the interval that is index of the starting frame.
fe : the upper endpoint of the interval that is index of the ending frame.
}

$ListIntervals$ **:** List of elements of Interval type.

Functions :

index getFs($Interval$) **:** gets the index of the starting frame fs of the interval $Interval$.

index getFe($Interval$) **:** gets the index of the ending frame fe of the interval $Interval$.

init($ListIntervals$) **:** initializes $ListIntervals$ as an empty list.

boolean isEmpty($ListIntervals$) **:** is a boolean function that returns true if the $ListIntervals$ is empty and false otherwise.

getFirst($ListIntervals$) **:** gets the first element $Interval$ of the $ListIntervals$.

addFirst(0, $Interval$, $ListIntervals$) **:** adds the element $Interval$ at the beginning of the list $ListIntervals$. The position zero refers to the first element of the list.

removeFirst($ListIntervals$) **:** removes the first element from the list $ListIntervals$.

sort($ListIntervals$) **:** sorts the elements of $ListIntervals$ in ascending order of their starting frame index fs.

TABLE 9.1 – Algorithm for the computation of the priority zone value

```
INPUT:
1) PriorList: list of the intervals (i.e domains) of the object frame
    sequences, which are related to all the objects appearing in the shot,
     and have a priority value equal to 1.

OUTPUT:
1) PZList: list of non-overlapping intervals elements, which is result
    from the union of the PriorList elements.

BEGIN
    //Initialize variables:
  init(PZList)

(1) sort(PriorList)
    q = getFirst(PriorList)
    removeFirst(PriorList)

(2) While ( isEmpty(PriorList) = false ) Do
      p = getFirst(PriorList)
      If getFs(p) \leq getFe(q) // note that getFs(q) < getFs(p) as the
          PriorList is ordered.
      /* This corresponds to Allen temporal relations (meets and overlaps
          ) between p and q. */
        If getFe(p) > getFe(q)
        /* We add the two ofs by extending the end frame of q to the end
            frame of p.*/
          fe(q) = fe(p)
        Else
          /* getFe(p) \leq getFe(q)
          * This corresponds to Allen temporal relations (equals,
          * starts, finishes, during) between p and q.
          * p can be ignored; we only need to remove it from PriorList
          * this will be done after the first If statement*/
        EndIf
      Else
      /* getFs(p) > getFe(q)
      * This corresponds to Allen temporal relation p takes place
      * after q. We add q to PZList, and restart the process from p. */
        addFirst(q, PZList)
        q = p
      EndIf
      removeFirst(PriorList)
    EndWhile

(3) addFirst(q, PZList)
    return PZList
END
```

9.2.3.3 Affected Priority Zone

Set of video intervals affected by an adaptation operation : given an adaptation operation $op \in \mathcal{OP}'$ and a shot $sh \in \mathcal{SH}$ to be adapted from an object $o \in \mathcal{O}$ in order to satisfy a user' semantic constraint $SemCt$, the set of video intervals affected by op is given by the function $affI : \mathcal{OP}' \times \mathcal{SH} \rightarrow \mathcal{P}(\mathcal{D}(I_f^v))$ whose codomain is the powerset of the set $\mathcal{D}(I_f^v)$. The function $affI(op, sh) \in \mathcal{P}(\mathcal{D}(I_f^v))$ is defined as follows :

$$affI(op, sh) = \begin{cases} I_f^{sh} & , \ if \ op = (Drop - Shot) \\ \bigcup_{i=1}^{|SOFS_o^{sh}|} I_f^{ofs_{o,i}^{sh}} & , \ if \ op = (Drop - SOFS) \\ \bigcup_{i=1}^{|SOFS_o^{sh}|} I_f^{ofs_{o,i}^{sh}} & , \ if \ op = (Drop - Object) \end{cases} \tag{9.12}$$

We refer the reader to Section 6 for the definition of the video domain I_{sh}^v and the set of all video intervals $\mathcal{D}(I_f^v)$.

The affected priority zone is the intersection between the priority zone in the shot and the set of video intervals affected by an adaptation operation. Given the interval set intersection operation $\bigodot$ (refer to Property 2), the affected priority zone apz is given by the following function :

$$\begin{array}{rccl} apz : & \mathcal{OP}' \times \mathcal{SH} & \longrightarrow & \mathcal{P}(\mathcal{D}(I_f^v)) \\ & (op, sh) & \longmapsto & apz(op, sh) = pz(sh) \bigodot affI(op, sh) \end{array} \tag{9.13}$$

Meanwhile, the codomain of the function apz is the powerset of the set $\mathcal{D}(I_f^v)$. Consequently, the priority zone $apz(op, sh)$ is in practice given as :

- op=(Drop-Shot)

$$apz(Drop - Shot, sh) = \begin{cases} I_f^{sh} \bigodot I_f^{sh} = I_f^{sh} & , \ if \ \rho_{sh} = 1 \\ \bigcup_{i=1}^{k} D_i \bigodot I_f^{sh} = \bigcup_{i=1}^{k} D_i & , \ if \ \rho_{sh} = 0 \end{cases} \tag{9.14}$$

In this case, we can therefore note that $apz(Drop - Shot, sh) = pz(sh)$.

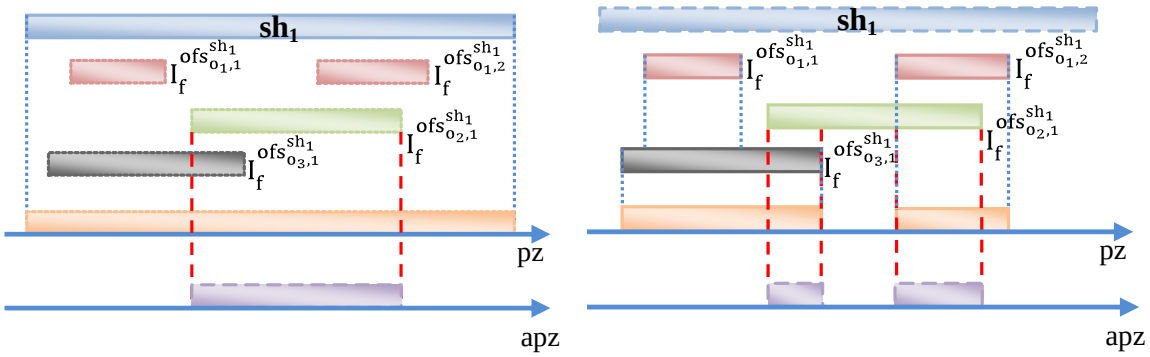
- op=(Drop-SOFS)

$$apz(Drop - SOFS, sh) = \begin{cases} I_f^{sh} \bigodot \bigcup_{i=1}^{|SOFS_o^{sh}|} I_f^{ofs_{o,i}^{sh}} = \bigcup_{i=1}^{|SOFS_o^{sh}|} I_f^{ofs_{o,i}^{sh}} & , \ if \ \rho_{sh} = 1 \\ \bigcup_{i=1}^{k} D_i \bigodot \bigcup_{i=1}^{|SOFS_o^{sh}|} I_f^{ofs_{o,i}^{sh}} = \bigcup_{i=1}^{l} C_i & , \ if \ \rho_{sh} = 0 \end{cases} \tag{9.15}$$

where l is the cardinality of the set of intervals C_i resulting from the intersection. Figure 9.5 illustrates the affected priority zone for the $Drop - SOFS$ operation on the object o_2.

- op=(Drop-Object)

For the $Drop - Object$ adaptation operation, the affected priority zone is equals to the one of the $Drop - SOFS$ as the zone subjected to the adaptation is the same for both operations.



(a) $\rho_{sh_1} = 1$ (b) $\rho_{sh_1} = 0$

FIGURE 9.5 – Calculating the affected priority zone for the $Drop - SOFS$ operation on o_3.

Example 15. Continuing our previous Example 14, let us compute the affected priority zone for each operation on the object o_2 (see Figure 9.5).
If $\rho_{sh_1} = 1$ then :
- $apz(Drop - Shot, sh) = [1, 120]$, and
- $apz(Drop - SOFS, sh) = apz(Drop - Object, sh) = [41, 95]$.
If $\rho_{sh_1} = 0$ then :
- $apz(Drop - Shot, sh) = \{[5, 55],\ [74, 100]\}$ which is the result of Algorithm 9.1, and
- $apz(Drop - SOFS, sh) = apz(Drop - Object, sh) = \{[41, 95]\} \bigodot \{[5, 55],\ [74, 100]\}$ $= \{[41, 55],\ [74, 95]\}$ which is calculated according to Property 2.

Based on the definition of the affected priority zone, we compute its size in pixels, denoted by $\gamma(apz(op, sh))$, as follows :

- op=(Drop-Shot)

$$\gamma(apz(Drop - Shot, sh)) = \delta(pz(sh)) \tag{9.16}$$

Such that $\delta(pz(sh))$ is the size of the priority zone in pixels (see Section 9.2.3.2).

- op=(Drop-SOFS)

$$\gamma(apz(Drop - SOFS, sh)) = \begin{cases} \sum_{i=1}^{|SOFS_o^{sh}|} length_{I_f^{ofs_{o,i}^{sh}}} \times size_f & , \ if\ \rho_{sh} = 1 \\ \sum_{i=1}^{l} length_{C_i} \times size_f & , \ if\ \rho_{sh} = 0 \end{cases} \tag{9.17}$$

where l is the cardinality of the set of intervals C_i resulting from the intersection (see Property 2).

- op=(Drop-Object)
Since the execution of the $Drop - Object$ operation yields a modification in the frame, thus the size of the affected priority zone in pixels, denoted by $\gamma(apz(Drop - Object, sh))$, is defined as the size of the object in each frame of the fs belonging to the intervals of the affected priority zone. Indeed, if the size of the object is close to the size of the frame, it is equivalent to dropping a frame. The affected priority zone $\gamma(apz(Drop - Object, sh))$ is defined as follows :

$$\gamma(apz(Drop - Object, sh)) = \begin{cases} \sum_{i=1}^{|SOFS_o^{sh}|} \sum_{j \in I_f^{ofs_{o,i}^{sh}}} size_{so^{f_j}} & , \ if \ \rho_{sh} = 1 \\ \sum_{i=1}^{l} \sum_{j \in C_i} size_{so^{f_j}} & , \ if \ \rho_{sh} = 0 \end{cases} \tag{9.18}$$

where for $\rho_{sh} = 1$, $size_{so^{f_j}}$ is the size of the spatial object in the frame f_j of the i^{th} $ofs_o^{sh} \in SOFS_o^{sh}$ within a shot sh, and $|SOFS_o^{sh}|$ is the cardinality of the set $SOFS_o^{sh}$. While for $\rho_{sh} = 0$, l is the cardinality of the set of intervals C_i resulting from the intersection between the priority zone and the zone affected by the $Drop - SOFS$ (see Equation 9.17), and $size_{so^{f_j}}$ is the size of the spatial object in the frame f_j of the i^{th} interval C within a shot sh.

In the rest of the thesis, we use the simplified notations δ, $\gamma_{Drop-Shot}$, $\gamma_{Drop-SOFS}$ and $\gamma_{Drop-Object}$ instead of $\delta(pz(sh))$, $\gamma(apz(Drop - Shot, sh))$, $\gamma(apz(Drop - SOFS, sh))$ and $\gamma(apz(Drop - Object, sh))$, respectively.

9.2.3.4 Affected Priority Area Ratio

As defined in Equation 9.9, the parameter p_2 measures the ratio of the exact size of the affected priority zone over the exact size of the priority zone in pixels. Among the different cases explained above, it is worthy to note that the priority zone can be equal to zero if $\rho_{sh} = 0$ and sh does not contain any prior object. Since no prior pixels are affected, we define p_2 to return 1 in that case. To summarize, given an object $o \in \mathcal{O}$, $\forall sh \in \mathcal{SH}$ and $op \in \mathcal{OP}'$ the general formula of the parameter p_2 is given as follows :

$$p_2(op, sh) = \begin{cases} 1 & , \ if \ \delta = 0 \ and \ \forall \ op \in \mathcal{OP}' \\ 0 & , \ if \ \delta \neq 0 \ and \ op = (Drop - Shot) \\ 1 - \frac{\gamma_{Drop-SOFS}}{\delta} & , \ if \ \delta \neq 0 \ and \ op = (Drop - SOFS) \\ 1 - \frac{\gamma_{Drop-Object}}{\delta} & , \ if \ \delta \neq 0 \ and \ op = (Drop - Object) \end{cases} \tag{9.19}$$

9.2.4 Perceived Quality

Temporal adaptation operations involving the removal of sequences of frames such as $Drop-Shot$ and $Drop-SOFS$, cause temporal gaps in the video content. Depending on their size, these gaps can produce undesired temporal impairments, thus minimizing the user's quality of experience. Indeed, a quality perception study conducted by the authors in [98] showed that people had a significant negative reaction to these temporal impairments (referred to as temporal discontinuities).

On the other hand, the spatial adaptation techniques involving the removal of the object in the frame and the implementation of the $Drop - Object$ operation (e.g., object inpainting), can also produce undesired artifacts in the frame. The major perceptually disturbing artifacts include damaged edges, blockiness, ringing, and especially blurriness. Indeed, Mahalingam analyzes the effect of inpainting on the human gaze, which describes the awareness with which an object can be viewed [86]. He succeeded in proving experimentally via eye tracking that gaze energy in the whole region of an inpainted image shows marked deviations from normal behavior correlated with the amount of inpainted artifacts. Since these inpainted artifacts are measured in the image, we refer to them as spatial artifacts in the rest of the thesis.

To this end, it is vital to develop functions that quantify the level of satisfaction of the user in the presence of temporal impairments and spatial artifacts. Recent neurophysiological research has found that spatial information (shape, size, etc.) and temporal information (motion, etc.) are processed separately in two visual pathways (ventral and dorsal streams) in the visual cortex of the human brain [43]. Based on this fact, the impact of the temporal impairments and the spatial artifacts on the quality of the user's visual perception are measured separately using the functions p_3 and p_4, respectively. Then, these two functions are combined together as parameters of the utility function, as stated in Section 9.2.

In order to define the p_3 (resp. p_4) functions, it is crucial to understand the quantitative relationship between the magnitude of the temporal gaps (resp. the magnitude of the spatial artifacts) and their impact on the quality of the user's visual perception. In psychophysics, the Weber-Fechner Law (WFL) is a fundamental law of sensory perception that expresses a logarithmic relationship between the physical magnitudes of stimuli (e.g., temporal gaps, spatial artifacts) and the perceived intensity of the stimuli. In fact, this law combines two different laws that we shortly described them below.

In 1834, the German physiologist Ernst Heinrich Weber was the first to discover that the minimum stimulus intensity variation necessary to produce a noticeable variation in perception between two stimuli, the so-called 'just-noticeable difference', is proportional to the magnitude of the initial stimulus [142]. Weber discovered this law in experiments on the thresholds of perception of lifted weights. The experiments show that if a person can distinguish between 20 g and 21 g of weight, then the increase for the just-noticeable difference for an initial stimulus of 40 g is 2 g. This law, known as Weber's Law, is expressed by : $\frac{dS}{S} = z$, where S is the initial intensity of stimulation, dS is the minimum noticeable variation and z is a constant known as the Weber fraction.

Later on, the philosopher and experimental psychologist Gustav Theodor Fechner extended Weber's Law to the known Weber-Fechner Law (WFL) [48]. He states that

the differential change in perception is proportional to the relative change of a physical stimulus :

$$dp = k \times \frac{dS}{S} \tag{9.20}$$

Here, dp is the differential change in perception, dS is the differential increase in the stimulus, S is the stimulus at the instant and a constant k is to be determined experimentally. Basically, WFL states that subjective sensation is proportional to the logarithm of the stimulus intensity. Indeed, an integration of the Equation 9.20 clearly shows the logarithmic relationship between stimulus and perception :

$$p = k \times \ln \frac{S}{S_0} \tag{9.21}$$

where p is the magnitude of the perception and S_0 is the threshold of stimulus below which it is not perceived at all.

This law turned out to be valid for almost all of our senses, like our auditory system (i.e., sound level is a logarithmic measure of the effective sound pressure of a sound relative to a reference value, and it is measured on a decibel scale), vision (i.e., stellar magnitude is measured in a logarithmic scale), and so forth.
Recent results from Quality of Experience (QoE) research succeeded to show that the relationship between user-perceived QoE and the size of a certain Quality of Service (QoS) parameter of the communication system also follows logarithmic laws [107] [51] [21]. Hinted at by the WFL, the authors in [107] express this QoE- QoS relationship in terms of a stimulus-perception, wherein the stimuli are seen as the technical parameters of the network (e.g., packet loss ratio), and the perception corresponds to the user's QoE.

Among these works, we point to the IQX Hypothesis described in [51]. In fact, the IQX Hypothesis is considered as an inversion of the WFL, and it was mathematically proven by the authors in [107] to be consistent with the fundamental observation of WFL. Put concisely, the IQX Hypothesis expresses the QoE as a function of impairment factors I_j, $1 \leq j \leq n$ corresponding to the QoS : $QoE = f(I_1,\ I_2,\ \ldots,\ I_n)$. Then, it derives the QoE function with a single impairment, while assuming that *the change of QoE depends on the current level of QoE - the expectation level - given the same amount of change of the QoS value* [51]. The authors illustrate this assumption by comparison with the quality of experience in a restaurant. If we dined in a five-star restaurant - and so had a high expectation level -, a small spot on the table cloth would strongly decrease the QoE. On the other hand, if the same thing occurred in a bar or pub - where the expectation level is lower - it would appear much less significant. Mathematically speaking, this assumption is defined as follows :

$$\frac{dQoE}{dQoS} = -b \times (QoE - c) \tag{9.22}$$

The solution of the differential equation 9.22 after integrating $dQoS$ is found to be an exponential decay, which is expressed by :

$$QoE = a \times exp(-b \times QoS) + c \tag{9.23}$$

where a is an initial value corresponding to the highest value of QoE in absence of impairment, b is a positive number, called the decay constant, to be defined experi-

mentally, and c is an additive constant corresponding to the lowest value of QoE when the impairment is so high.

Given this explanation, it sounds very logical to say that the relationship between the stimulus, i.e., impairments resulting from the spatial and temporal adaptation operations, and their impact on the quality of the user's visual perception is non-linear. For the purpose of this thesis, we study the impact of the size of the temporal gaps and the size of the spatial artifacts in p_3 and p_4, respectively. Though we are aware of the usefulness of the saliency maps and motion maps in objectively assessing the perceptual quality of a spatio-temporally adapted video, their investigation is out of the scope of this thesis. Moreover, despite the fact that the exponential and logarithmic models are consistent with the observation of WFL, making a decision about how to model p_3 and p_4 still needs to be done. In the remainder of this section, we argue the choice of exponential decay model for p_3 and p_4 based on the IQX Hypothesis.

9.2.4.1 Temporal Perceived Quality

The purpose of the temporal perceived quality parameter p_3 is to quantify the level of satisfaction of the user in the presence of temporal impairments. It is defined as a non-linear function of a single parameter factor i.e., $p_3 = f(GapSR)$, where $GapSR$ is the ratio of the gaps size in a shot produced by an adaptation operation. For $Drop-Object$, the value of $GapSR$ is equal to 0, since $Drop-Object$ does not cause temporal impairments. Yet, for $Drop-Shot$ and $Drop-SOFS$ operations, $GapSR$ is computed by the affected area ratio function (refer to Section 9.2.1), where a value of 1 means that whole frames of the shot were removed. Furthermore, the parameter p_3 is a decreasing function over its entire domain $[0, 1]$ (refer to Section 9.2.1). Indeed, the value of p_3 is equal to 1 when $GapSR = 0$, and it approaches 0 when $GapSR = 1$.

As mentioned before, in order to decide whether p_3 is modeled as an exponential or logarithmic function, we need to understand the change of the temporal perceived quality dp_3 with respect to the change of the gap size ratio $dGapSR$. Basically, if we are adapting a shot sh_i of high priority $\rho_{sh_i} = 1$, a temporal gap of a small size is easily perceived by the user, and decreases p_3. On the other hand, a temporal gap with the same size appearing in a shot sh_j of low priority $\rho_{sh_j} = 0$ with $size_{sh_i} = size_{sh_j}$, would not have the same effect on the user's perception. In this context, we adopt the assumption of the IQX Hypothesis, since the temporal perceived quality change dp_3 for a given fixed change of the gap size ratio $dGapSR$ is proportional to the current level of the perception p_3. Hence, the parameter p_3 is modeled as an exponential decay (see Figure 9.6), and is defined as follows :

$$p_3 = a \times exp(-b \times GapSR) + c \tag{9.24}$$

where a is an initial value corresponding to the value of p_3 when $GapSR = 0$ (i.e., a is equal to 1 as the temporal perceived quality is at the highest value $p_3 = 1$ when $GapSR = 0$), b is a positive number called the decay constant, to be defined experimentally, and the value of the additive constant c approaches 0 as p_3 is approximately 0 when $GapSR$ approaches 1.

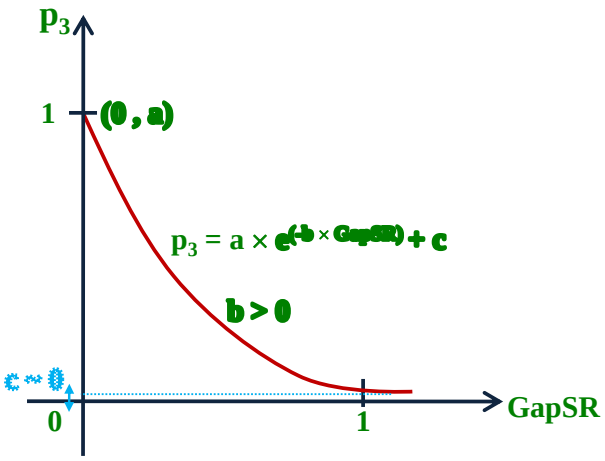


FIGURE 9.6 – The parameter p_3 defined as an exponential decay function.

Consequently, we define $p_3 : \mathcal{OP}' \times \mathcal{SH} \rightarrow [0,1]$ as follows :

$$p_3(op, sh) = exp(-b \times aar(op, sh)) \tag{9.25}$$

Accordingly, for each adaptation operation, the function $p_3(op, sh)$ is expressed by :

$$p_3(op, sh) = \begin{cases} exp(-b \times aar(Drop - Shot, sh)) & , \quad if\ op = (Drop - Shot) \\ exp(-b \times aar(Drop - SOFS, sh)) & , \quad if\ op = (Drop - SOFS) \\ 1 & , \quad if op = (Drop - Object) \end{cases} \tag{9.26}$$

In Section 13.4.1, subjective assessments demonstrate that the parameter p_3 is a good predictor of the user's QoE for a decay constant value b, estimated to $2,871$. Indeed, the Pearson linear correlation coefficient between the prediction value of p_3 and the Mean Opinion Score (MOS) ratings of the human observer, shows the high accuracy (i.e., 0.93) of the model.

9.2.4.2 Spatial Perceived Quality

The purpose of the spatial perceived quality parameter p_4 is to quantify the level of satisfaction of the user in the presence of spatial artifacts produced by the inpainting algorithms. Several aspects play a role in the quality of inpainting. For instance, some existing inpainting algorithms strive to remove an object from the frame by filling the hole using a smoothing process. This process tends to blur the regions so that the boundary region is implicitly handled. Moreover, an earlier work in analyzing error inpainting [18] shows that the quality of the inpainting depends more on the shape of the image inpainting domain than on the size or total area of the inpainting domain. Actually, the inpainting of a narrow shape, such as text covering a frame, cannot be seen as the inpainting of a bottle from the frame. However, we cannot generalize these results, especially because we are usually dealing with large rather than narrow objects. Therefore, we can only assume that the inpainting process for a given spatial object so^f of an object o in a frame f, will produce artifacts of size $(size_{art})$. The value of

$size_{art}$ is more or less large, depending on the inpainting algorithm, so that in the worst case it will be equal to the size of the spatial object $size_{sof}$ in pixels (refer to Section 7.3.1). For instance, Figure 9.7 illustrates the result of inpainting two objects of different size. As can be observed from the figure, the artifacts are obvious in the right image. This is due to the size of the object in one hand, and its heterogeneous texture on a heterogeneous background on the other hand.

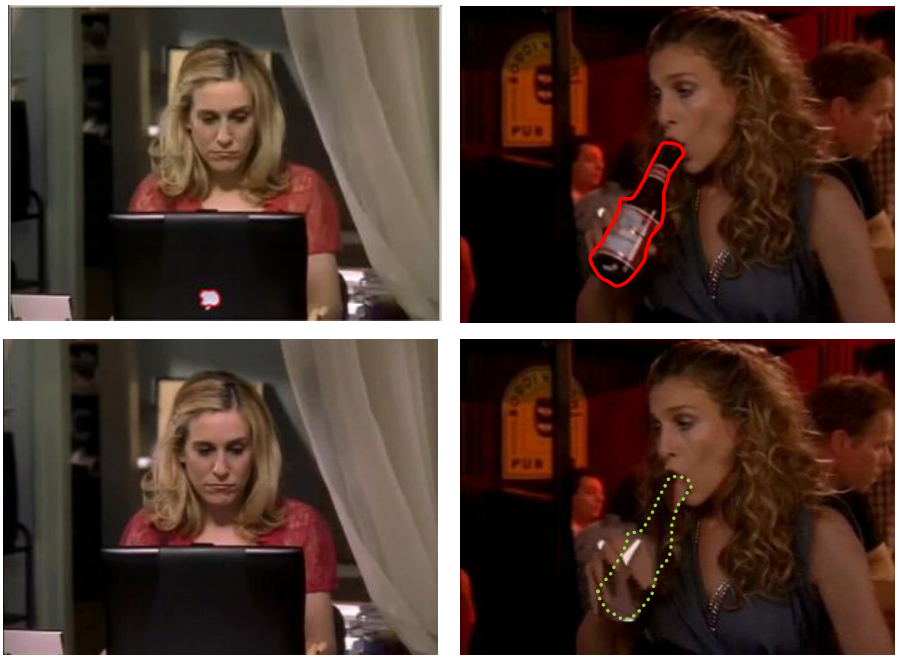


FIGURE 9.7 – Effect of inpainting two objects of different size and texture.

Accordingly, the parameter p_4 is defined as a non-linear function of a single parameter factor i.e., $p_4 = f(ArtSR)$, whereas $ArtSR$ is the ratio of the spatial artifacts size in a shot produced by an adaptation operation. For $Drop-Shot$ and $Drop-SOFS$, the value of $ArtSR$ is set to 0, since these operations do not cause spatial artifacts. Yet, for the $Drop-Object$ operation, $ArtSR$ is defined as a function $ArtSR: \mathcal{O}\times\mathcal{SH} \rightarrow [0,1]$ such that $\forall\ sh \in \mathcal{SH},\ o \in \mathcal{O}$, $ArtSR(o, sh)$ is the sum of the spatial artifact ratio of each object frame (see Section 7.3.2), and is expressed by :

$$ArtSR(o,sh) = \frac{\sum_{i=1}^{|SOFS_o^{sh}|} \sum_{j \in I_f^{ofs_{o,i}^{sh}}} \frac{size_{art_j}}{size_{f_j}}}{\sum_{i=1}^{|SOFS_o^{sh}|} length_{ofs_{o,i}^{sh}}} \tag{9.27}$$

where $\frac{size_{art_j}}{size_{f_j}}$ is the artifact ratio caused by inpainting the spatial object so^{f_j} related to o in a specific frame f_j of the i^{th} $ofs_{o,i}^{sh} \in |SOFS_o^{sh}|$ within the shot sh.

Furthermore, the parameter p_4 is a decreasing function over its entire domain $[0,1]$ (refer to Section 9.2.1). Indeed, the value of p_4 is equal to 1 when $ArtSR = 0$, and it approaches 0 when $ArtSR = 1$. Similar to p_3, the parameter p_4 is modeled as an exponential decay and defined as follows :

$$p_4 = a' \times exp(-b' \times ArtSR) + c' \tag{9.28}$$

where a' is an initial value corresponding to the value of p_4 when $ArtSR = 0$ (i.e., a' is equal to 1 as the spatial perceived quality is at the highest value $p_4 = 1$ when $ArtSR = 0$), b' is a positive number called the decay constant, to be defined experimentally, and the value of the additive constant c' approaches 0 as p_4 is approximately 0 when $ArtSR$ approaches 1.

Consequently, we define $p_4 : \mathcal{OP}' \times \mathcal{SH} \rightarrow [0, 1]$ as follows :

$$p_4(op, sh) = exp(-b' \times ArtSR(o, sh)) \tag{9.29}$$

Accordingly, for each adaptation operation, the function $p_4(op, sh)$ is expressed by :

$$p_4(op, sh) = \begin{cases} 1 & , \quad if op = (Drop - Shot) \\ 1 & , \quad if op = (Drop - SOFS) \\ exp(-b' \times ArtSR(o, sh)) & , \quad if\ op = (Drop - Object) \end{cases} \tag{9.30}$$

9.2.5 Processing Cost

The processing cost parameter aims to model the adaptation execution time for the three operations $Drop - Shot$, $Drop - SOFS$, and $Drop - Object$. In order to predict the processing costs of adapting a shot sh containing an object o, we have to describe the adaptation execution chain for each adaptation operation.

$Drop{-}Shot$ refers to cutting a sequence of frames from the video. While $Drop{-}Shot$ can mostly be executed in the compressed domain, dropping sequences of frames may raise the need to re-encode some frames. This occurs when the frames in the shot are referenced by other frames outside the shot. Therefore, the execution time of this operation is the time of cutting the shot and re-encoding the remaining frames.

$Drop - SOFS$ refers to removing the object frame sequences that are related to a specific object in the shot. Similar to $Drop - Shot$, the execution of the $Drop - SOFS$ could require re-encoding some of the frames. Thus, the execution time of this operation is the time of cutting $|size_{ofs_o^{sh}}|$ object frame sequences and re-encoding the remaining frames.

Finally, $Drop - Object$ refers to removing the spatial object related to the object o from each frame within the shot. Instead of decoding the whole shot, the execution of this operation begins with cutting the ofs_o^{sh}, and then decoding it. Afterwards, every spatial object related to the object o is removed from each frame of the ofs_o^{sh}. Finally, the adapted frames are re-encoded.

Based on the explanation above, we need to profile the execution time of encoding/-decoding algorithms and adaptation tools. Basically, a video is a three-dimensional array of color pixels. Two dimensions serve as spatial direction of the moving pictures, and one dimension represents the time. A video frame is defined as a set of all pixels at a specific time, and a temporal segment is defined as a frame sequence (e.g., $shot$, ofs, $SOFS$).

Video encoding uses spatial and temporal redundancy to encode a video. Spatial redundancy can be used by intraframe compression of the information representing a frame. Temporal redundancy can be used by interframe compression which refers to temporal encoding that uses earlier and/or later frame(s) to compress a frame. Many video codecs have been proposed. Benchmarks generally measure their performance by the number of frames per second (fps) they can encode/decode. Thus, for our processing cost parameter, we assume that the used codec has been tested so that this value is known and can be used to measure the processing cost of encoding or decoding a given number of frames.

Besides the video codecs, adaptation tools are also needed to perform the temporal adaptation by removing frame sequences ($shot$, ofs) from the video, and spatial adaptation by removing objects from the frames. Among numerous cutting tools and inpainting algorithms [50] [115] in the existing literature, we choose Avidemux [11] for temporal adaptation and the inpainting algorithm described in [95] for spatial adaptation. Using these tools, we have implemented our adaptation operators, and studied their performance with varying object size and number of frames. Annex F provides detailed information about the evaluation of the execution time of each adaptation operator. From the experimental results, we conclude that the cutting time is constant with respect to the number and length of the frame sequences to be removed.

Obviously, the execution time for the cutting, decoding, encoding and inpainting will vary from one adaptation tool to another. Therefore, we characterize an adaptation tool by a profile defined as follows :

- Adaptation Tool Profile :
  - t_{cut} refers to the time (ms) for cutting a sequence of frames.
  - t_{inp} refers to the time (ms) for replacing one pixel of the object with one neighborhood pixel.

- Codec Profile :
  - t_{enc} refers to the time (ms) for encoding one frame.
  - t_{dec} refers to the time (ms) for decoding one frame.

Accordingly, given a shot sh consisting of object frame sequences $ofs_o^{sh} \in SOFS_o^{sh}$ containing an object o to be removed, the execution time of each adaptation operations is given as follows :

- **op = Drop-Shot :**
  - $t_{cut} + (t_{enc} \times \alpha)$, such that α is the number of frames that need to be re-encoded due to the cut of the shot.

- **op = Drop-SOFS :**
  - $t_{cut} + (t_{enc} \times \beta)$, such that β is the number of frames that need to be re-encoded due to the cut of the ofs_o^{sh}.

- **op = Drop-Object :**
  - t_{cut} is the execution time of cutting $|SOFS_o^{sh}|$ object frame sequences ofs_o^{sh}.

11. http://www.avidemux.org/

- $t_{dec} \times \sum_{i=1}^{|SOFS_o^{sh}|} length_{ofs_{o,i}^{sh}}$ is the execution time of decoding the frames of every ofs_o^{sh}.
- $\sum_{i=1}^{|SOFS_o^{sh}|} \sum_{j \in I_f^{ofs_{o,i}^{sh}}} t_{inp}(size_{so^{f_j}})$ is the execution time of removing the spatial object related to o from each frame of every ofs_o^{sh}.
- $t_{enc} \times (\psi + \sum_{i=1}^{|SOFS_o^{sh}|} length_{ofs_{o,i}^{sh}})$, such that ψ is the number of frames that need to be re-encoded due to the adaptation of the ofs_o^{sh}.

It is summarized as follows :

$$\begin{cases} t_{cut} \ + \ (t_{enc} \times \alpha) & , \ if \ op = (Drop - Shot) \\ t_{cut} \ + \ (t_{enc} \times \beta) & , \ if \ op = (Drop - SOFS) \\ t_{cut} \ + \ t_{dec} \times \sum_{i=1}^{|SOFS_o^{sh}|} length_{ofs_{o,i}^{sh}} & \\ + \sum_{i=1}^{|SOFS_o^{sh}|} \sum_{j \in I_f^{ofs_{o,i}^{sh}}} t_{inp}(size_{so^{f_j}}) & \\ + \ t_{enc} \times \ (\psi + \sum_{i=1}^{|SOFS_o^{sh}|} length_{ofs_{o,i}^{sh}}) & , \ if \ op = (Drop - Object) \end{cases} \tag{9.31}$$

Moreover, based on the experimental results in Annex F, empirical observation shows that the execution time of the $Drop-Shot$ and $Drop-SOFS$ is negligible with respect to the execution time of $Drop-Object$. Indeed, even for the simplest case where the color and texture characteristics of object and background are homogeneous, and the size of the spatial object is too small (i.e., 0.25% of the image size), the execution time of the $Drop - Object$ is twice that of a $Drop - Shot$ and $Drop - SOFS$.

In order to normalize the processing cost p_5, we divide the execution time of each operation by the execution time of the most expensive operation, $Drop-Object$. Thus, we define the function $p_5 : \ \mathcal{OP}' \times \mathcal{SH} \rightarrow \ [0,1]$ such that $\forall \ sh \in \mathcal{SH}$, $p_5(op, sh)$ is given as follows :

$$p_5(op, sh) = \ 1 - \frac{execution \ time \ of \ the \ operation}{execution \ time \ of \ the \ Drop - Object} \tag{9.32}$$

Thus :

$$p_5(op, sh) = \begin{cases} 1 - \dfrac{t_{cut} \ + \ (t_{enc} \times \alpha)}{execution\ time\ of\ Drop - Object} & , \ if\ op = (Drop - Shot) \\ 1 - \dfrac{t_{cut} \ + \ (t_{enc} \times \beta)}{execution\ time\ of\ Drop - Object} & , \ if\ op = (Drop - SOFS) \\ 0 & , \ if\ op = (Drop - Object) \end{cases} \tag{9.33}$$

9.2.6 Utility Function Output Format

Given a scene $sc \in \mathcal{SC}$ and an object $o \in \mathcal{O}$ to be removed, the output of Utility Function (UF) can be described by a matrix $U \in [0,\ 1]^{|\mathcal{OP}'| \times\ |Adp\mathcal{SH}_{sc,o}|}$, which contains a row for each adaptation operator $op \in \mathcal{OP}'$ and a column for each shot $sh \in Adp\mathcal{SH}_{sc,o}$. An element $u_{i,j} = UF(op_i,\ sh_{\tau_{sc,o}(j)})$ of the matrix U is the utility value of applying an adaptation operator over a shot (refer to Section 8.2.2). Formally, the matrix U is defined as follows :

$$U : \begin{cases} \{1, \ldots, |\mathcal{OP}'|\} \times \{1, \ldots, |Adp\mathcal{SH}_{sc,o}|\} & \longrightarrow & [0,\ 1] \\ (i,\ j) & \longmapsto & u_{i,j} \ = \ UF(op_i,\ sh_{\tau_{sc,o}(j)}) \end{cases} \tag{9.34}$$

where $|\mathcal{OP}'| = 3$ (i.e., $op_1 = (Drop - Object),\ op_2 = (Drop - SOFS)$ *and* $op_3 = (Drop - Shot)$), and $|Adp\mathcal{SH}_{sc,o}|$ is the number of shots in the scene sc that are to be adapted (refer to Section 8.2.2).

Example 16. Let us consider $Adp\mathcal{SH}_{sc_1,o_2} = \{sh_1,\ sh_4,\ sh_7,\ sh_8\ \}$, where $sc_1 \in \mathcal{SC}$ and $o_2 \in \mathcal{O}$ is the object to be removed. Figure 9.8 illustrates the utility matrix U such that all the elements $u_{i,j}$ are similarly calculated as $u_{1,1}$.

$$\begin{array}{c} \textbf{Mat. U} \\ \textbf{op}_\textbf{1} \\ \textbf{op}_\textbf{2} \\ \textbf{op}_\textbf{3} \end{array} \begin{array}{c} \begin{array}{cccc} (\textbf{sh}_\textbf{1})_\textbf{1} & (\textbf{sh}_\textbf{4})_\textbf{2} & (\textbf{sh}_\textbf{7})_\textbf{3} & (\textbf{sh}_\textbf{8})_\textbf{4} \end{array} \\ \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{1,4} \\ u_{2,1} & u_{2,2} & u_{2,3} & u_{2,4} \\ u_{3,1} & u_{3,2} & u_{3,3} & u_{3,4} \end{bmatrix} \end{array}$$

$$\begin{array}{c} \textbf{Mat. U} \\ \textbf{op}_\textbf{1} \\ \textbf{op}_\textbf{2} \\ \textbf{op}_\textbf{3} \end{array} \begin{array}{c} \begin{array}{cccc} \textbf{sh}_\textbf{1} & \textbf{sh}_\textbf{4} & \textbf{sh}_\textbf{7} & \textbf{sh}_\textbf{8} \end{array} \\ \begin{bmatrix} 0,50 & 0,80 & 0,75 & 0,34 \\ 0,65 & 1,00 & 0,90 & 0,67 \\ 0,70 & 0,40 & 0,20 & 0,80 \end{bmatrix} \end{array}$$

$$u_{1,1} = UF(op_1,\ sh_1) = \sum_{i=1}^{5} (w_i \times p_i(op_1,\ sh_1))$$

FIGURE 9.8 – Illustration of a utility matrix U.

9.3 Synthesis

In this chapter, we have presented the semantic object-based video adaptation process, which lays at the core of the PIAF framework, the ADTE. We also discussed the selection of the best adaptation plan that maximizes the QoE of an adapted video content. In particular, we tackle the problem of estimating the impact of applying different types of adaptation operations to a shot in order to satisfy user constraints.

To this end, we defined a Utility Function (UF), which integrates semantic concerns with users' perceptual consideration. This function evaluates the utility of an adaptation according to five parameters with a threefold purpose : 1) preserving the semantic integrity of the content by minimizing the overall impact of the adaptation especially on semantically critical parts of the content ; 2) maximizing the spatial and temporal perceived quality of the adapted content ; and 3) minimizing the processing cost of the adaptation operation.

Given a scene with shots to be adapted, both UF and its parameters were described in detail, along with their related functions. The output of the UF is a utility matrix. Based on this matrix, we explain in the next chapter how the ADTE performs a multi-level piecewise reasoning to compute the adaptation plan that maximizes the user perceived quality, with and without the presence of owner constraints.

Chapter 10
Adaptation Plan Computation

In this chapter, we describe how an adaptation plan is computed. The execution of the plan must provide an adapted scene that satisfies the end-user constraints as well as the owner constraints that may apply. Given the utility matrix and the owner constraints as an input, the problem is defined as assigning an adaptation operator to each shot such that the resulting adaptation plan maximizes the global utility and satisfies the owner and end-user constraints. In some cases, satisfying the end-user constraints may violate the owner's intellectual property rights, thus resulting in conflicting constraints. This situation leads to an optimization problem, which we map to 0-1 Multiple-Choice Knapsack Problem (0-1 MCKP).

10.1 Adaptation Plan Specification

Given a specific pair $(sc, o) \in \mathcal{SC} \times \mathcal{O}$, the definition of an adaptation plan consists of assigning an adaptation operator $op \in \mathcal{OP}'$ for each shot $sh \in Adp\mathcal{SH}_{sc,o}$ (refer to Section 8.2.2). In the remainder of this section, we present the formal definition of the assign function followed by the definition of the adaptation plan and its global utility.

Definition 39. Assign function *assign* **:** Given a specific pair $(sc, o) \in \mathcal{SC} \times \mathcal{O}$, the choice of an adaptation operator $op \in \mathcal{OP}'$ for a shot $sh \in Adp\mathcal{SH}_{sc,o}$ is formally described by the function *assign* defined as :

$$\begin{array}{rccl} assign_{sc,o} & : & Adp\mathcal{SH}_{sc,o} & \longrightarrow & \mathcal{OP}' \\ & & sh & \longmapsto & op = assign_{sc,o}(sh) \end{array} \tag{10.1}$$

10.1.1 Formal Definition of an Adaptation Plan

An adaptation plan for a specific pair $(sc, o) \in \mathcal{SC} \times \mathcal{O}$ **:** Let $ap(sc, o)$ denote an adaptation plan for a specific pair $(sc, o) \in \mathcal{SC} \times \mathcal{O}$. We define $ap(sc, o)$ as an ordered $|Adp\mathcal{SH}_{sc,o}|-$tuple of pairs $(op, sh)_j \in \mathcal{OP}' \times Adp\mathcal{SH}_{sc,o}$, such that for each shot $sh_{\tau_{sc,o}(j)} \in Adp\mathcal{SH}_{sc,o}$, only one pair $(op = assign_{sc,o}(sh_{\tau_{sc,o}(j)}), sh_{\tau_{sc,o}(j)}) \in \mathcal{OP}' \times$

$\{sh_{\tau_{sc,o}(j)}\}$ exists. Therefore, we define $ap(sc, o)$ as follows :

$$\begin{array}{c} ap(sc,o) = (\ (op,sh)_1,\ \ldots,\ (op,sh)_{|Adp\mathcal{SH}_{sc,o}|}\ ) \\ such\ that\ (op, sh_{\tau_{sc,o}(j)})_j \in \mathcal{OP}' \times \{sh_{\tau_{sc,o}(j)}\}\ \ for\ all\ 1 \leq j \leq |Adp\mathcal{SH}_{sc,o}|. \end{array} \tag{10.2}$$

We refer the reader to Section 8.2.2 for the definition of the $Adp\mathcal{SH}_{sc,o}$ set and the τ function.

Example 17. Let us consider $Adp\mathcal{SH}_{sc_1,o_2} = \{sh_1,\ sh_4,\ sh_7,\ sh_8\ \}$, such that $o_2 \in \mathcal{O}$ is the object to be removed from the scene $sc_1 \in \mathcal{SC}$. A possible adaptation plan for (sc_1, o_2) could be the following :

$$\begin{array}{rl} ap(sc_1,o_2) & = (\ (op_1,\ sh_{\tau_{sc_1,o_2}(1)})_1, (op_2,\ sh_{\tau_{sc_1,o_2}(2)})_2, (op_3,\ sh_{\tau_{sc_1,o_2}(3)})_3, (op_2,\ sh_{\tau_{sc_1,o_2}(4)})_4\ ) \\ & = (\ (op_1,\ sh_1)_1,\ (op_2,\ sh_4)_2,\ (op_3,\ sh_7)_3,\ (op_2,\ sh_8)_4\ ). \end{array}$$

The set of all adaptation plans for a specific pair $(sc, o) \in \mathcal{SC} \times \mathcal{O}$: Let $Ap(sc, o)$ denote the set of all adaptation plans for a specific pair $(sc, o) \in \mathcal{SC} \times \mathcal{O}$. It corresponds to the set of all possible operator assignments for the shots in $Adp\mathcal{SH}_{sc,o}$:

$$\begin{array}{rl} Ap(sc,o) = & \times_{j=1}^{|Adp\mathcal{SH}_{sc,o}|} \mathcal{OP}' \times \{sh_{\tau_{sc,o}(j)}\}, such\ that \\ & sh_{\tau_{sc,o}(j)} \in Adp\mathcal{SH}_{sc,o}\ for\ all\ 1 \leq j \leq |Adp\mathcal{SH}_{sc,o}| \end{array} \tag{10.3}$$

The set of all adaptation plans for all pairs $(sc, o) \in \mathcal{SC} \times \mathcal{O}$: Based on Equation 10.3, we denote by $\bigcup_{(sc,o)\ \in\ \mathcal{SC}\times\mathcal{O}} Ap(sc, o)$ the set of all adaptation plans for all pairs $(sc, o) \in \mathcal{SC} \times \mathcal{O}$ such that :

$$\bigcup_{(sc,o)\ \in\ \mathcal{SC}\times\mathcal{O}} Ap(sc,o) \ = \ \bigcup_{(sc,o)\ \in\ \mathcal{SC}\times\mathcal{O}} \times_{j=1}^{|Adp\mathcal{SH}_{sc,o}|} \mathcal{OP}' \times \{sh_{\tau_{sc,o}(j)}\}$$

Therefore, given a pair $(sc, o) \in \mathcal{SC} \times \mathcal{O}$, the adaptation plan can be formally described as the result of the function ap from $\mathcal{SC} \times \mathcal{O}$ to $\bigcup_{(sc,o)\ \in\ \mathcal{SC}\times\mathcal{O}} \times_{j=1}^{|Adp\mathcal{SH}_{sc,o}|} \mathcal{OP}' \times \{sh_{\tau_{sc,o}(j)}\}$. We define the function ap as follows :

$$\begin{array}{rcl} ap\ :\ \mathcal{SC} \times \mathcal{O} & \longrightarrow & \bigcup_{(sc,o)\ \in\ \mathcal{SC}\times\mathcal{O}} \times_{j=1}^{|Adp\mathcal{SH}_{sc,o}|} \mathcal{OP}' \times \{sh_{\tau_{sc,o}(j)}\} \\ (sc,o) & \longmapsto & ap(sc,o) = (\ (op,sh)_1,\ \ldots,\ (op,sh)_{|Adp\mathcal{SH}_{sc,o}|}\ ) \end{array} \tag{10.4}$$

such that $ap(sc, o)$ is an ordered $|Adp\mathcal{SH}_{sc,o}|-$tuples of pairs $(op, sh_{\tau_{sc,o}(j)})_j \in \mathcal{OP}' \times \{sh_{\tau_{sc,o}(j)}\}\ for\ all\ 1 \leq j \leq |Adp\mathcal{SH}_{sc,o}|$.

10.1.2 Output Format of an Adaptation Plan

Let $A \in \{0,\ 1\}^{|\mathcal{OP}'| \times\ |Adp\mathcal{SH}_{sc,o}|}$ denote a logical matrix, which contains a row for each adaptation operator $op \in \mathcal{OP}'$ and a column for each shot $sh \in Adp\mathcal{SH}_{sc,o}$. An element $a_{i,j}$ of the matrix A may take two values : 1 if the adaptation operator op_i

is assigned to the shot $sh_{\tau_{sc,o}(j)}$ and 0 otherwise. Formally, the matrix A is defined as follows :

$$A : \begin{cases} \{1,\ \ldots,\ |\mathcal{OP}'|\} \times \{1,\ \ldots,\ |Adp\mathcal{SH}_{sc,o}|\} \longrightarrow \{0,\ 1\} \\ (i,\ j) \longmapsto a_{i,j} = \begin{cases} 1 & ,\ if\ op_i = assign(sh_{\tau_{sc,o}(j)}) \\ 0 & ,\ if\ op_i \neq assign(sh_{\tau_{sc,o}(j)}) \end{cases} \end{cases} \tag{10.5}$$

Therefore, the choice of an adaptation plan can be described by A iff $\sum_{i=1}^{|\mathcal{OP}'|} a_{i,j} = 1,\ for\ all\ 1 \leq j \leq |Adp\mathcal{SH}_{sc,o}|$. This corresponds to the choice of one single operation per shot. For instance, Figure 10.1 illustrates the adaptation plan selected in example 17. We recall that op_1, op_2 and op_3, respectively, corresponds to the $Drop - Object$, $Drop - SOFS$ and $Drop - Shot$.

Matrix U	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1 = Drop - Object}$	0,50	0,80	0,75	0,34
$\mathbf{op_2 = Drop - SOFS}$	0,65	1,00	0,90	0,67
$\mathbf{op_3 = Drop - Shot}$	0,70	0,40	0,20	0,80

Matrix A	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1 = Drop - Object}$	1	0	0	0
$\mathbf{op_2 = Drop - SOFS}$	0	1	0	1
$\mathbf{op_3 = Drop - Shot}$	0	0	1	0

$$\mathbf{ap(sc_1, o_2)} = (\ (op_1,\ sh_1)_1,\ (op_2,\ sh_4)_2,\ (op_3,\ sh_7)_3,\ (op_2,\ sh_8)_4\ )$$

FIGURE 10.1 – Adaptation plan described by the logical matrix A.

10.1.3 Global Utility of an Adaptation Plan

Let $GUF(ap(sc,o))$ denote the global utility of an adaptation plan $ap(sc,o)$. We define $GUF(ap(sc,o))$ as the sum of the utility value $u_{i,j}$ of each pair $(op,sh)_j$ of $ap(sc,o)$ multiplied by the size of the shot $size_{sh_{\tau_{sc,o}(j)}}$.

$$GUF(ap(sc,o)) = \sum_{j=1}^{|Adp\mathcal{SH}_{sc,o}|} \sum_{i=1}^{|\mathcal{OP}'|} u_{i,j} \times a_{ij} \times\ size_{sh_{\tau_{sc,o}(j)}} \tag{10.6}$$

Given this definition, we consider that the shots are adapted independently of each other, and their adaptation does not influence one another.

10.2 Adaptation Plan Computation

In this section, we explain how the best adaptation plan is computed without and with owner constraints, respectively. In the absence of owner constraints, this computation simply consists of assigning the operator of highest utility value to each shot.

The plan resulting from this computation is called best adaptation plan. However, the problem is significantly more complex when dealing with owner constraints, particularly if the computed best adaptation plan conflicts with the owner constraints. We show that in this case, the number of feasible adaptation plans grows exponentially, which makes an exhaustive search inapplicable. Consequently, we formulate the adaptation plan computation problem as an optimization problem. The originality of this contribution is in mapping the adaptation plan computation problem to the 0-1 Multiple-Choice Knapsack Problem (0-1 MCKP).

10.2.1 Adaptation Plan Selection without Owner Constraints

As described in Section 9.2, the domain of the utility for an adapted shot is $[0, 1]$ where the higher the value, the higher the utility. Meanwhile, at the scene level, the best adaptation plan is the one that maximizes the global utility value. Thus, based on Equation 10.6, we can deduce that the plan with the highest $GUF(ap(sc, o))$ is the one which assigns the operator with the highest utility value to each shot undergoing the adaptation.

Formally, let $bap(sc, o)$ denote the best adaptation plan of a specific pair $(sc, o) \in \mathcal{SC} \times \mathcal{O}$. Hence, $bap(sc, o)$ is defined as an $ap(sc, o)$ such that for all $1 \leq j \leq |Adp\mathcal{SH}_{sc,o}|$, only the pairs $(op, sh_{\tau_{sc,o}(j)}) \in \mathcal{OP}' \times \{sh_{\tau_{sc,o}(j)}\}$ with maximum utility values $UF(op, sh_{\tau_{sc,o}(j)}) = \max\limits_{op \in \mathcal{OP}'} \{UF(op, sh_{\tau_{sc,o}(j)})\}$ are selected. We define the function $bap(sc, o)$ as follows :

$$
\begin{aligned}
bap \ : \ \mathcal{SC} \times \mathcal{O} \ &\longrightarrow \ \bigcup_{(sc,o) \ \in \ \mathcal{SC} \times \mathcal{O}} \times_{j=1}^{|Adp\mathcal{SH}_{sc,o}|} \mathcal{OP}' \times \{sh_{\tau_{sc,o}(j)}\} \\
(sc, o) \ &\longmapsto \ bap(sc, o) = (\ (op, sh)_1, \ \ldots, \ (op, sh)_{|Adp\mathcal{SH}_{sc,o}|}\ ) \\
&\quad\ \ such\ that\ for\ all\ 1 \leq j \leq |Adp\mathcal{SH}_{sc,o}| \\
&\quad\ \ (op, sh_{\tau_{sc,o}(j)}) : UF(op, sh_{\tau_{sc,o}(j)}) = \max_{op \in \mathcal{OP}'} \{UF(op, sh_{\tau_{sc,o}(j)})\}
\end{aligned}
\tag{10.7}
$$

Back to example 17, Figure 10.2 illustrates the computation of the best adaptation plan $bap(sc_1, o_2)$ in the absence of the owner constraints.

Once selected, the plan $bap(sc_1, o_2)$ can be directly sent to the adaptation execution engine. However, in the presence of owner constraints and if the $bap(sc, o)$ violates one of them, a different plan must be computed following the process described in the next section.

10.2.2 Adaptation Plan Selection with Owner Constraints

The selection method of an adaptation plan in presence of owner constraints depends on the type of constraints imposed by the owner. Clearly, if the video cannot be adapted, no adaptation plan can be computed. If specific shots cannot be adapted or

$$\begin{array}{c} \textbf{Matrix U} \\ \textbf{op}_1 \\ \textbf{op}_2 \\ \textbf{op}_3 \end{array} \begin{array}{c} \begin{array}{cccc} \textbf{sh}_1 & \textbf{sh}_4 & \textbf{sh}_7 & \textbf{sh}_8 \end{array} \\ \begin{bmatrix} 0,50 & 0,80 & 0,75 & 0,34 \\ 0,65 & \boxed{1,00} & \boxed{0,90} & 0,67 \\ \boxed{0,70} & 0,40 & 0,20 & \boxed{0,80} \end{bmatrix} \end{array}$$

$$\begin{array}{c} \textbf{Matrix A} \\ \textbf{op}_1 \\ \textbf{op}_2 \\ \textbf{op}_3 \end{array} \begin{array}{c} \begin{array}{cccc} \textbf{sh}_1 & \textbf{sh}_4 & \textbf{sh}_7 & \textbf{sh}_8 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \boxed{1} & \boxed{1} & 0 \\ \boxed{1} & 0 & 0 & \boxed{1} \end{bmatrix} \end{array}$$

$$\textbf{bap}(\textbf{sc}_1, \textbf{o}_2) = (\ (op_3, sh_1),\ (op_2, sh_4),\ (op_2, sh_7),\ (op_3, sh_8)\ )$$

FIGURE 10.2 – Best adaptation plan computation in the absence of owner constraints.

removed, the problem is similar to the one exposed in Section 10.2.1, except that some operator assignments are not available. However, if the owner constraint specifies a percentage of the number of the frames that can be dropped, the problem can be more complex. Indeed, if the best adaptation plan $bap(sc, o)$ violates this constraint, then another feasible plan close to the optimal feasible adaptation plan must be found. An exhaustive search to identify this optimal feasible plan is not an option as the number of possible plans increases exponentially with the number of shots. For instance, from Figure 10.2, it is easy to see that for a given set of shots in a scene sc and a set of adaptation operators $\mathcal{OP}'$, there are $|\mathcal{OP}'|^{|Adp\mathcal{SH}_{sc,o}|}$ possible adaptation plans. In the rest of this chapter, we describe the method for resolving this type of constraint.

To begin with, we need to consider the total number of dropped frames for an adaptation plan. This number should not exceed the maximum number of frames, denoted by $maxfnb$, allowed by the owner constraint. To this end, we define a size matrix $S \in \mathbb{N}^{|\mathcal{OP}'| \times\ |Adp\mathcal{SH}_{sc,o}|}$ such that each element $s_{i,j}$ corresponds to the number of dropped frames while executing the adaptation operator op_i over the shot sh_j. It is worthy to note that $s_{3,j}$ for the operation $op_3 = Drop-Shot$ corresponds to the length of the shot $sh_{\tau_{sc,o}(j)}$. The matrix S is therefore defined as follows :

$$S : \left\{ \begin{array}{rcl} \{1, \ldots, |\mathcal{OP}'|\} \times \{1, \ldots, |Adp\mathcal{SH}_{sc,o}|\} & \longrightarrow & \mathbb{N} \\ (i,\ j) & \longmapsto & s_{i,j} \end{array} \right. \tag{10.8}$$

For instance, Figure 10.3 illustrates an example of the matrix S for the previously described example 17. Let us consider the owner constraint : "No more than 10% of the scenes can be removed". Assuming that the size of the scene sc_1 is 2880 frames, then the number of dropped frames should not exceed $maxfnb = 2880 * 0,1 = 288$. As depicted in the figure, the choice of the best adaptation plan $bap(sc_1, o_2)$ violates the owner constraint as it yields a removal of 458 frames. Thus, another plan should be chosen to reduce the removal to at least $458 - 288 = 170$ frames.

10.2.3 Mapping the Adaptation plan computation to 0-1 MCKP

We formulate the adaptation plan computation problem as a *0-1 Multiple-Choice Knapsack Problem* (0-1 MCKP) [68]. The MCKP, also known as Knapsack Problem

Matrix U	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1}$	0,50	0,80	0,75	0,34
$\mathbf{op_2}$	0,65	1,00	0,90	0,67
$\mathbf{op_3}$	0,70	0,40	0,20	0,80

Matrix S	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1}$	0	0	0	0
$\mathbf{op_2}$	68	40	90	20
$\mathbf{op_3}$	148	155	160	180

$$\mathbf{bap(sc_1, o_2)} = (\ (op_3, sh_1),\ (op_2, sh_4),\ (op_2, sh_7),\ (op_3, sh_8)\ )$$

FIGURE 10.3 – Example of a matrix U with its matrix S.

with *Generalized Upper Bound* (GUB) *Constraints*, is a 0-1 knapsack problem formulated as follows : given k classes $C_1, \ldots, C_k$ of the item set C and a *weight* and a *profit* value attributed to each item, choose one item from each class such that the *profit sum* is maximized, and the *weight sum* of the chosen items does not exceed the *capacity* of the knapsack (see Figure 10.4).

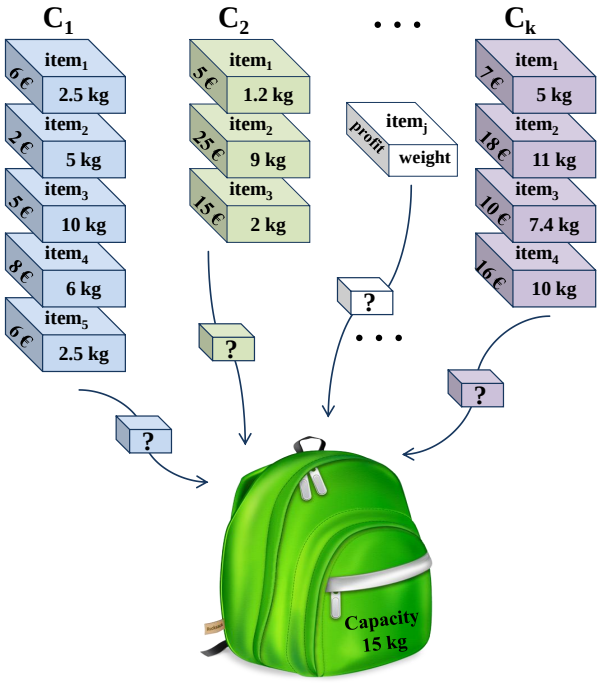


FIGURE 10.4 – The Multiple-Choice Knapsack Problem (MCKP).

Accordingly, since in our case one and only one operation can be chosen for each shot, we map the optimal adaptation plan computation problem to 0-1 MCKP (see Figure 10.5). We consider $|Adp\mathcal{SH}_{sc,o}|$ classes $C_1, \ldots, C_{|Adp\mathcal{SH}_{sc,o}|}$ of items to choose for the optimal adaptation plan, such that each class corresponds to a column $col_j = (a_{1,j}, \ldots, a_{|Adp\mathcal{SH}_{sc,o}|,j})$ in the logical matrix A (refer to Section 10.1.2). Thus, an item $i \in C_j$ is none other than a pair $(op_i, sh_{\tau_{sc,o}(j)}) \in \mathcal{OP}' \times Adp\mathcal{SH}_{sc,o}$. Moreover, each item has a utility $u_{i,j} \in U$ and a size $s_{i,j} \in S$, respectively, corresponding to the *weight* and *profit* value in the MCKP.

Hence, given the owner constraint stating that not more than $maxfnb$ frames of the scene sc can be dropped, the objective is to choose one adaptation operation for each shot in order to maximize the global utility of the adaptation plan, and to avoid that the *sum of dropped frames* exceeds $maxfnb$. Therefore, the optimal adaptation

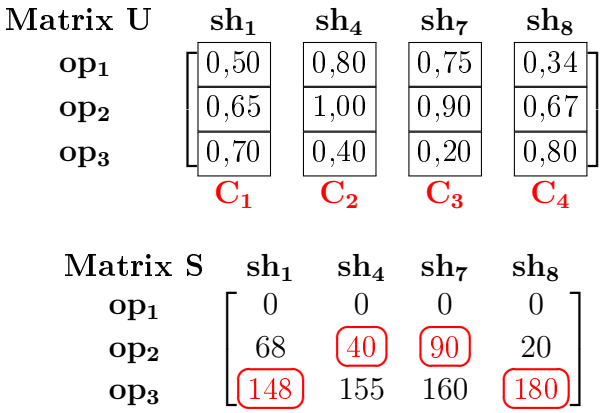


FIGURE 10.5 – Mapping the Adaptation plan computation to 0-1 MCKP.

plan computation problem may be formulated as follows :

$$
\begin{aligned}
&\textit{maximize} && \sum_{j=1}^{|AdpSH_{sc,o}|} \sum_{i \in C_j} u_{i,j} \times a_{i,j} && \\
&\textit{subject to} && \sum_{j=1}^{|AdpSH_{sc,o}|} \sum_{i \in C_j} s_{i,j} \times a_{i,j} \leq maxfnb, && \\
& && \sum_{i \in C_j} a_{i,j} = 1, && j = 1, \ldots, |AdpSH_{sc,o}| \\
& && a_{i,j} \in \{0,1\}, && j = 1, \ldots, |AdpSH_{sc,o}|, \\
& && && i \in C_j = \{1, \ldots, |OP'|\}
\end{aligned}
\tag{10.9}
$$

Such that all the coefficients $u_{i,j}$, $s_{i,j}$ and $maxfnb$ are non-negative values, and the classes are disjoint $C_h \cap C_k = \emptyset$ *for all* $h \neq k$. Each class corresponds to a column in the matrix A, therefore all classes have the same size $|OP'|$, and the total number of items to consider is $|OP'| \times |AdpSH_{sc,o}|$.

The 0-1 MCKP has been intensively studied in the literature. It is NP-hard as it contains Knapsack Problem (KP) as a special case, but it can be solved in pseudo-polynomial time through dynamic programming [68] [29]. Many algorithms, such as the dynamic programming method, the enumerative method, the branch-and-bound method and the heuristic algorithms are proposed for solving the MCKP [30] [29] [101]. A survey of these solutions can be found in [68]. For the implementation of our framework PIAF, we adopted a slight modification of the greedy algorithm approach proposed by Pisinger in [101]. We evaluated experimentally this algorithm to verify that it runs in a reasonable execution time and generates adaptation plans which quality is close to the one of the optimal plan. Further details regarding this algorithm and its evaluation are given in Annex E.

10.3 Synthesis

In this chapter, we formally described the computation of an adaptation plan with and without owner constraints. In the absence of owner constraints, we selected the adaptation plan that has the maximum global utility and satisfies the end-user constraints. However, in the presence of owner constraints, we explained that the selection method of an adaptation plan depends on the type of constraints imposed by the owner. We particularly showed that in case of an owner constraint specifying a percentage of the number of the frames that can be dropped, and in case satisfying the end-user constraint will violate this owner constraint, the adaptation plan computation problem lead to an optimization problem. To this end, we proposed an original solution by formulating the adaptation plan computation problem as a 0-1 MCKP.

Part III

Implementation of PIAF

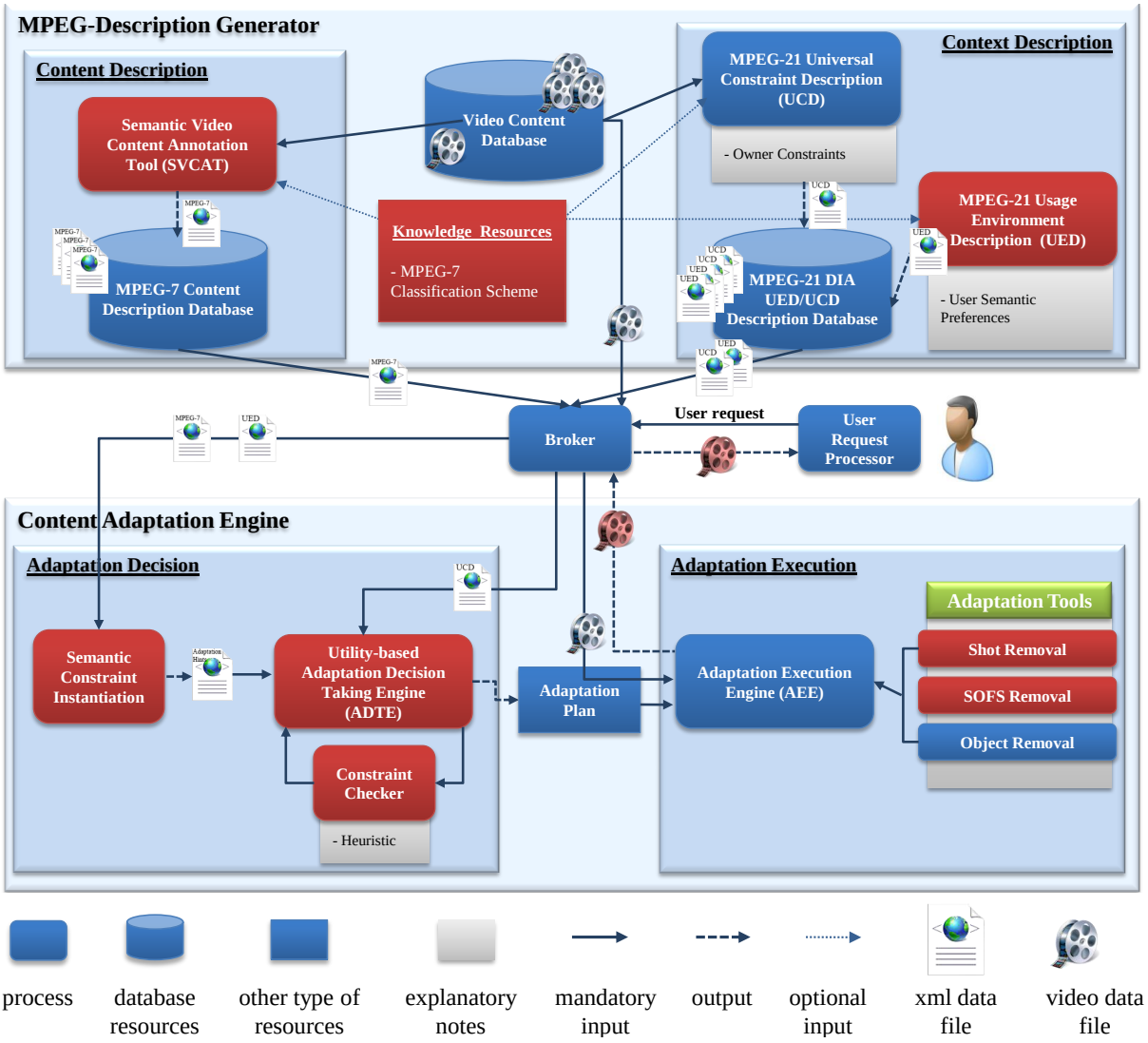


FIGURE 10.6 – PIAF architecture.

This part discusses the detailed implementation of PIAF architecture, which is depicted in Figure 10.6. This architecture is an implementation of the generic MPEG-based architecture described in Chapter 5.

The red modules in the figure are the ones that have been implemented in this thesis, and are described in the remaining chapters of this part. For the *Context Description Module*, we developed Semantic Video Content Annotation Tool (SVCAT) to generate accurate and MPEG-7 based video annotations, which are compliant with the video model proposed in Chapter 7. Regarding the *Content Description Module*, we implemented the user profile using the MPEG-7 and MPEG-21 description tools. An extension to the MPEG-7 and MPEG-21 standards was also proposed to enable the representation of users semantic constraints compliant to the model presented in Chapter 8. Moreover, the two processes of the *Adaptation Decision Module* were also implemented : 1) the semantic instantiation constraint process takes UED and MPEG-7 content descriptions as input, and generates the *InstantiatedConstraint* as output ; 2) the latter along with the UCD serve as input for the Adaptation Decision Taking Engine (ADTE), which generates an adaptation plan as output. This generation is done with the support of the constraint checker, which verifies that the plan does not violate the owner intellectual property rights. In case of violation, it executes the heuristic to find the feasible plan. It is worthy to underline the fact that the *Adapta-*

tion Decision Module is fully automated. This is possible thanks to the availability of standardized semantic descriptions for both video content and user constraints, which highly contribute to the automation of the filtering and adaptation process without any human intervention. The implementation of the ADTE and the constraint checker is a straightforward transcription of the algorithms presented in Chapter 9 for the UF and Annex E for the heuristic. Therefore, they are not detailed in this part.

Although the tools related to the $Drop - Shot$ and the $Drop - SOF$ operations were implemented, the implementation of the adaptation tools used by the ADTE is out of scope of this thesis. Therefore, the subjective evaluation tests were done on pre-adapted videos, instead of doing the adaptation online (see Chapter 13).

Chapter 11

SVCAT : Semantic Video Content Annotation Tool

In this Chapter, we present our Semantic Video Content Annotation Tool (SVCAT), which assists annotators in generating video annotations compliant with the video model proposed in Chapter 7. To begin, we discuss the requirements of a video content annotation tool in the context of PIAF and briefly overview existing video annotation tools according to these requirements. A positioning of SVCAT among these tools is also discussed. Afterwards, we analyze the requirement specifications for developing a video annotation tool. Then, we present the architecture of SVCAT and describe in details the functionalities of each of its module. In order to justify our design choices for each part of the SVCAT prototype, analytical and experimental evaluations of the some existing approaches are also presented. Moreover, we describe the experimental evaluation regarding the accuracy and performance of SVCAT. We conclude this chapter with a synthesis of SVCAT.

11.1 Requirements of PIAF Video Content Annotation Tool

As stated in the previous chapters, PIAF is an MPEG standard compliant and object-based adaptation framework preserving the user perceived quality. Moreover, in order to realize an efficient adaptation in PIAF, the availability of a rich, accurate and MPEG-7 based video annotation is a fundamental requirement. Therefore, the annotation tool for PIAF must fulfil the following requirements :

1. **Video model :** the aim of PIAF is to realize a semantic object-based adaptation. A video model was developed in Chapter 7, which consists of all the semantic, structural (i.e., scene, shot) and spatial (i.e., object) information that are required to assist the adaptation process in PIAF. Thus, the annotation tool should provide video annotation according to this model.

2. **MPEG-7 metadata format :** a metadata file format is required to concretely represent the information of the video model. As mentioned in Chapter 5, PIAF is an MPEG standard compliant framework. Thus, an MPEG-7 metadata format is required to enable the integration and exploitation of the generated metadata, by the various modules of PIAF. By doing so, we will alleviate the problem of interoperability between the PIAF modules.

3. **Object selection accuracy :** PIAF aims to adapt the video content at the object level while preserving the quality perceived by the user. As discussed in Section 9.2.4 of Chapter 9, this quality is measured with respect to the artifacts produced by the inpainting algorithm used by the $Drop - Object$ operator. The input of this inpainting algorithm is the annotation about the spatial object. Therefore, the accuracy of the selection of the spatial object at the level of the annotation tool is a key factor in the quality of the adaptation result.

4. **Degree of automation :** semantic and structural annotation of videos includes several processes that are very time consuming if they have to be executed manually. This is in particular the case for identifying the temporal structure and localizing the spatial object in each frame. Therefore, to facilitate the generation of rich annotation, the degree of automation of the tool should be as high as possible.

In the literature, several annotation tools have been developed to describe video content. An overview and comparison of these tools can be found in the following survey by Dasiopoulou et al. [25]. However, these available tools does not cover all the requirements needed for the purpose of PIAF. Therefore, we developed the Semantic Video Content Annotation Tool (SVCAT) [34]. To outline the need for SVCAT, we now briefly discuss other tools from the literature and position SVCAT among them. The comparison is done with respect to the aforementioned requirements for the purpose of PIAF. As depicted in the Table 11.1, most tools use self-defined XML formats for output descriptions, thus complicating the integration of the produced metadata in PIAF. Only SVAS [65] and VideoAnnEx [121] follow the MPEG-7 standard providing a good exchangeability and compatibility of the produced metadata.

Regarding the video segmentation, very few tools can automatically identify temporal segments and provide them with a semantic description (e.g., frame start, length). Only Advene [80], SVAS and VideoAnnex can perform automatic shot detection. With regards to scene segmentation, none of the tools so far provide automatic scene detection. Indeed, there has not been an efficient approach until now that works reliably in a non-restricted domain.

Concerning the spatial localization of objects, the situation is quite similar. Only VideoAnnEx, VIA [1] and SVAS allow selection and annotation of objects within the video. Nevertheless, the propagation of the object description and its spatial properties over consecutive frames still requires human intervention, and is deemed to be manual. For instance, the propagation of the object description is done either by dragging it while the video is playing (i.e., VIA), or by copying it with one mouse click to detected similar regions in the consecutive frames (i.e., SVAS). Moreover, none of these tools supports an automatic propagation of the object's spatial properties, such that its contour/boundary is accurately tracked in the video while generating in parallel the description of its shape and size. VIA and VideoAnnex represent objects using bounding rectangles. SVAS provides a slightly higher degree of precision and uses bounding polygons to represent the objects contour.

To position SVCAT among these tools, we analyze it regarding these aforementioned requirements, which are of utmost importance for an object-based adaptation. The positioning is presented in the last row of Table 11.1. Compared to the presented tools, SVCAT provides significant advantages with respect to interoperability issues, accuracy

Tool	Metadata Format	Localization temporal structure		Localization spatial structure		
		automatic shot	automatic scene	object annotation	selection accuracy	propagation of description
VIA [1]	XML	✖	✖	✔	rectangular bounding box	manual
Ontolog [47]	RDF	✖	✖	✖	-	-
VideoAnnEx [121]	MPEG-7	✔	✖	✔	rectangular bounding box	manual
Advene [80]	custom XML	✔	✖	✖	-	-
Elan [74]	custom XML	✖	✖	✖	-	-
Anvil [70]	custom XML	✖	✖	✖	-	-
SVAS [65]	MPEG-7	✔	✖	✔	polygon region	manual
SVCAT [34]	MPEG-7	✔	✖	✔	exact region	automatic

Table 11.1 – Positioning of SVCAT among current video annotation tools.

of the object representation and degree of automation.

11.2 Specifications of Video Content Annotation Tool

In this section, we examine the main requirements for an efficient video content annotation tool that enables the production of structural and semantic metadata at different levels of granularity. These requirements are considered in the light of the criteria discussed in the previous section : interoperability, accuracy and high degree of automation. We first describe the segmentation process, which partitions the video into temporal segments, i.e., scenes and shots. We outline the fact that human intervention in this process is unavoidable, especially in defining scenes. Then, a synthesis of the video content analysis phase, namely the generation of metadata related to objects and their spatial properties, is presented. To this end, we overview some of the existing object representation and selection algorithms used by the annotator to provide annotation of the spatial object within a frame (refer to Section 7.3). Moreover, we survey the object tracking algorithms that take the spatial object selected by the annotator as an input, and returns the sets of similar spatial objects in different frames as an output. Finally, we explore the need for an expressive model to capture all the extracted metadata using a standardized metadata format.

11.2.1 Video segmentation

Segmentation consists of decomposing video content in temporal units, usually shots and scenes. The problem of automatic shot boundary detection has attracted much attention, enabling state-of-the-art shot segmentation techniques to reach satisfying levels of performance, as demonstrated by the TREC Video Retrieval Evaluation track

(TRECVID) [119]. Despite a number of promising proposals, automatic scene detection on the other hand remains an open research issue [116]. We argue that a productive tool should limit itself to incorporating a shot detection technique, and include functions for manually correcting the detected shots and grouping them into scenes.

11.2.2 Video analysis

In the following, we consider the most challenging task of video analysis, which is the object annotation.

11.2.2.1 Spatial Object Representation

In order to annotate the spatial properties of video objects, a representation model must first be selected. Several models have been proposed in the literature. A survey of the most common representation models could be found in [146]. As depicted in Figure 11.1, the spatial object within the frame can be represented as sets of points, simple geometric shapes (e.g., rectangle, ellipse), or by using articulated shape models, skeletal models and contour or silhouette representations.

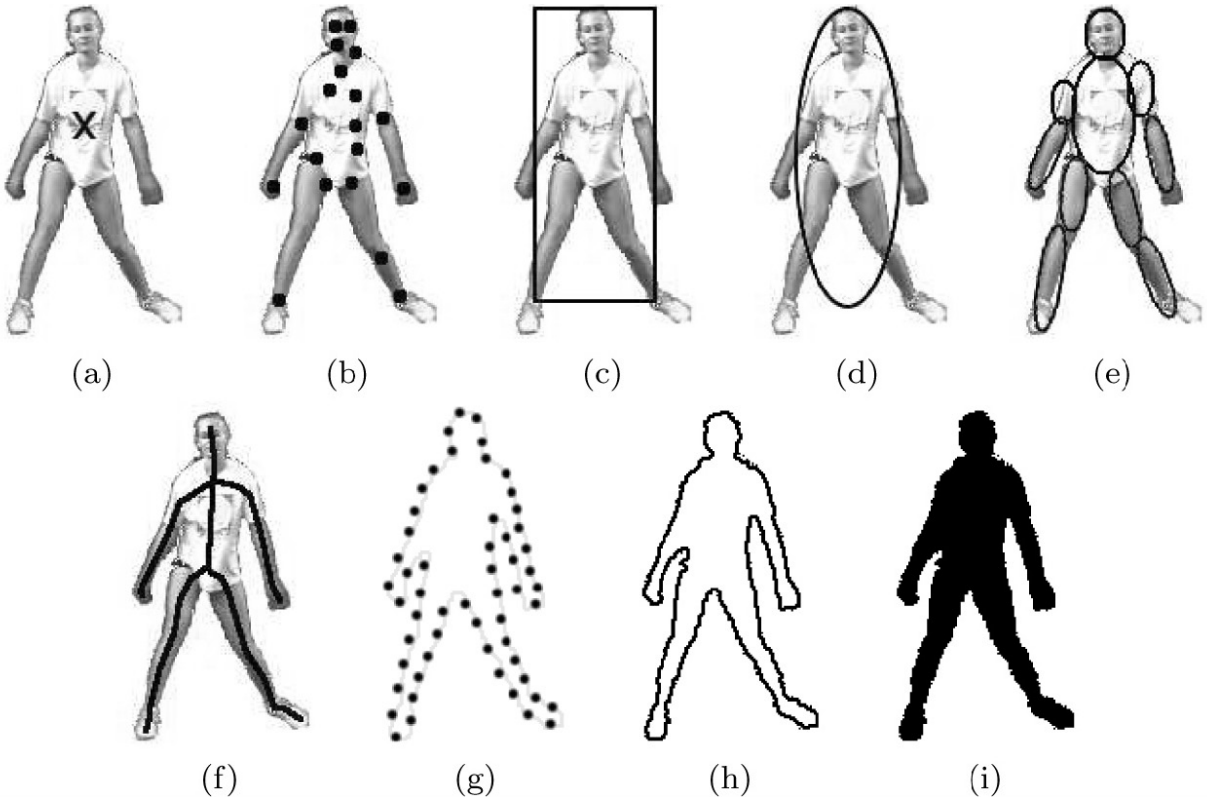


FIGURE 11.1 – Object representations. (a) centroid, (b) multiple points, (c) rectangular patch, (d) elliptical patch, (e) part-based multiple patches, (f) object skeleton, (g) control points on object contour, (h) complete object contour, (i) object silhouette (c.f. [146])

The contour corresponds to the set of pixels forming the boundary of an object, whereas the silhouette is the region inside the contour. For the purpose of SVCAT, we choose a combination of both contour and silhouette representation model. The former

is the most appropriate one to fulfil our goal of having an accurate representation, which implies an exact size of the spatial object in pixel within each frame. Whereas the latter facilitates the computation of low-level features over the whole object. This representation also has the advantage of being able to support a huge set of object deformations, which facilitates the representation of complex, non-rigid objects (e.g., pedestrians) at pixel accuracy.

Such a combined representation can be implemented using level sets [97]. The level set method uses a closed curve Γ in the two dimensional space to represent the contour. Γ is implicitly represented using an auxiliary function $\varphi : R^2 \times R \rightarrow R^1$ on a fixed Cartesian grid. This function is called the level set function. The values of φ are the euclidean distances from the contour Γ, which is represented by the zero level set of φ.

$$\Gamma = \{(x, y) \,|\, \varphi(x, y) = 0\} \tag{11.1}$$

The inside of the region delimited by Γ (i.e., silhouette) is given negative values $\varphi(x) < 0$ and the outside of the region positive values $\varphi(x) > 0$. The level set methodology provides some nice features. Unlike other representations (e.g., splines), it can handle topological changes in the object appearance like, for instance, the splitting and merging of regions. Additionally, intrinsic geometrical properties can be derived directly from the level set.

11.2.2.2 Object Selection

Object selection approaches for video annotation tools may be categorized into fully automatic, semi-automatic and manual methods. Automatic selection approaches are based on machine learning (e.g., supervised learning as in Adaptive Boosting [77]) and require the use of training data. This has the disadvantage of restricting the application to a specific domain for which prior data is available, and means that only the types of objects that appear in the provided data may be discovered. On the other hand, manual object selection is a very tedious task, which is also prone to errors. Thus, a tool that restricts itself to these methods cannot be realistically used to annotate large video collections. Therefore, we argue that a semi-automatic approach is best suited. The idea is that the annotator provides an initial region selection, and that an image segmentation uses it as an input to automatically compute an exact contour.

Image segmentation has been extensively researched in the image processing community. Three major techniques have emerged in recent years : Mean-Shift-Clustering [23], segmentation based on graph cuts, like the GrowCut algorithm [137] and active contour techniques, also known as snakes [66]. Mean-Shift clustering is not suited to our purposes. Indeed, it is highly dependent on the number of regions for segmentation, resulting frequently in over- or under- segmentation compared to human perception of objects. In a graph cut approach, the user labels a number of pixels either as belonging to the object or the background. Based on this input, the algorithm iteratively assigns labels to all other pixels of the image. In order to decide whether a pixel belongs to object or background, the method examines its similarities to neighbouring pixels that have already been labelled. The process is repeated until all pixels have been processed. An active contour approach starts from a first selection of the contour of the object provided by the user. The algorithm expands this contour line until it tightly encloses the intended contour. Contour evolution is governed by minimizing an energy functional.

Both graph cuts and active contour approaches are appropriate for our requirements. Thus, we have conducted experiments in order to evaluate their performance in the context of our tool (see section 11.3.2.1), from which we concluded that an active contour approach is the best choice.

11.2.2.3 Object Tracking

Though an object selection functionality as described above facilitates the exact definition of the spatial object corresponding to an object in a frame, doing so in each frame in which an object appears is a very cumbersome task. In order to automate this process, a tracking approach can be used to re-detect the object in all subsequent frames based on an initial selection provided by the annotator. Many object tracking methods have been proposed in the literature [146]. To select an appropriate approach, the following requirements must be considered :

1. **Object representation :** from the requirements regarding object representation detailed above, we can conclude that the object tracker should make use of a silhouette or contour for object representation.

2. **User input :** the initial spatial object selection provided by the annotator can be considered as reliable. The tracker should be able to make use of this information as much as possible and require no further user input.

3. **Generic algorithm :** objects may have various characteristics (different motion, non-rigid, complex shapes, etc.). The tracker should be generic enough to cope with all these types of objects. Moreover, the tracker should not introduce further constraints on the properties of the object (features, motion speed and degree of similarity of objects between frames).

4. **No assumption of previous data :** To keep the tool generic, the selected tracker cannot rely on any other previous data than the initial selection of the object region, such as training data in which objects have been identified.

Due to the first requirement, we restrict ourselves to silhouette trackers, excluding trackers that use other object representations. This category comprises shape-matching and contour evolution approaches. Shape-matching approaches try to iteratively match a representation of the object in each consecutive frame. They are not appropriate in our case because they cannot deal with non-rigid objects. The second category of approaches comprises two sub-categories, based either on state space models or on direct minimization of an energy functional. State Space models define a model of the object's state, containing shape and motion parameters of the contour. Tracking is achieved by updating this model so that the posterior probability of the contour is maximized. This probability depends on the model state in the current frame and on a likelihood describing the distance of the contour from observed edges. Direct minimization techniques implement tracking by trying to evolve an initial contour in each frame until a contour energy function is minimized. Methods in this category differ in their minimization method (greedy method or gradient descent) and their contour energy function, which is defined with respect to temporal information either

by means of a temporal gradient (optical flow) or of appearance statistics computed from the object and the background.

Approaches based on state space models require training data, and thus are not appropriate. Many direct minimization approaches of the literature have to be excluded as well because they are not generic enough and require training data or additional user input. This led us to narrow our study to the tracking methods proposed by Yilmaz et al. [147] and by Shi and Karl [114].

Yilmaz et al. evolve the contour using color and texture features within a band around the object's contour. Through this band, they aim to combine region-based and boundary-based contour tracking methods in a single approach. Objects are represented by level sets. Tracking of multiple objects is possible as well. A disadvantage of the algorithm is its explicit handling of occlusion. Even if an object is occluded by another object, its position is estimated by the tracker. This means that the tracker would calculate a contour for an object even if it is not visible and hence this region would be annotated as an object region erroneously.

Similar to this approach, Shi and Karl propose a tracking method based on a novel implementation of the level set representation and the idea of region competition [148]. They use color and texture information to model object and background region. Contour evolution is achieved by applying simple operations on the level set, like switching elements between lists. Switching decisions are obtained by estimating the likelihood of pixels around the zero level set to belong to a particular region (region competition). The approach requires no training and uses a simple tracking model, which computes the contour of the object in the current frame based on the information from the last frame. It can be extended to track multiple objects.

We conclude this analysis by opting for Shi and Karl's method, which combines satisfactory tracking accuracy with sound performance and does not have problems with occlusion. The implementation of this approach in our tool is described in section 11.3.2.2.

11.2.3 Semantic annotation

The abstraction levels for metadata associated with a video resource may range from low-level features to high-level semantic information. While structural metadata can be derived (semi-)automatically from the low-level features, the extraction of descriptive metadata that refers to the high level features and is based on knowledge, still require user intervention. This is due to the well-known problem of the semantic gap. Clearly, the representation of this knowledge affects the automation of annotation. Thus, the crucial need of an annotation tool that can support the annotator to increase the efficiency of the manual process. More, to alleviate the problem of interoperability, the annotation vocabulary should be chosen from a specific controlled vocabulary.

11.2.4 Video model and metadata format

In order to achieve interoperable and machine understandable annotations, there is a need to formalize and well define the semantics of the annotation vocabulary. To organize this information, the tool must base on a video annotation model. To enable fine-grained video annotation, this model must be expressive, especially in order to properly link the semantic descriptions and the structural elements of the video. Moreover, the model must be implemented in a metadata file format. In this regard, to make the tool interoperable, it is appropriate to use the multimedia content description standard MPEG-7 and a predefined controlled vocabulary, using for instance MPEG classification schemes.

11.3 Architecture and Functionalities of SVCAT

Figure 11.2 illustrates the conceptual architecture of SVCAT, which we implement for the purpose of PIAF. To begin, the annotator has to load a video and a MPEG-7 compliant Classification Schemes (CS). Indeed, a priority CS and object terms CS have to be specified for semantically annotating the temporal segments and video objects, respectively. After this initialization phase, the actual annotation process starts with the video segmentation and description of the temporal structure. This process is followed by a localization and annotation of the information related to the object : semantic, associated spatial objects in each frame, and the shot object frame sequences in the shots.

In the remainder of this section, we describe the functionalities of each of its modules and argue our design choices. For a step-by-step example of a video content annotation using SVCAT, we refer the reader to Annex B.

11.3.1 Temporal Structure Localization and Annotation

As depicted in Figure 11.2, the temporal structure localization module consists of two steps : shot boundary detection and scene construction. SVCAT benefits from VAnalyzer [124], which performs an automatic detection of the shot boundaries based on various techniques (e.g., Canny Edge Detector or motion compensation). More Information about VAnalyzer can be found in [124]. Figure 11.3 illustrates the Graphical User Interface (GUI) of VAnalyzer with a video being played out for shot detection.

Once the shots are detected, a user interface for the manual post-processing of the shot detection appears, as depicted in Figure 11.4. This interface represents an overview of the result of the shot detection algorithm. Each shot is represented by its first and its last frame. Based on this result, the next step is to compose the scenes by manually connecting adjacent shots. Indeed, the *scene connector button* represented by a double arrow symbol between the shots, is used to connect the shots into scenes. It is worth to note that the scene construction was implemented in the context of SVCAT. At this stage, each shot is considered as a scene by default. Thus, the number of scene is equal to the number of the detected shots. Then, by clicking on the scene connector button,

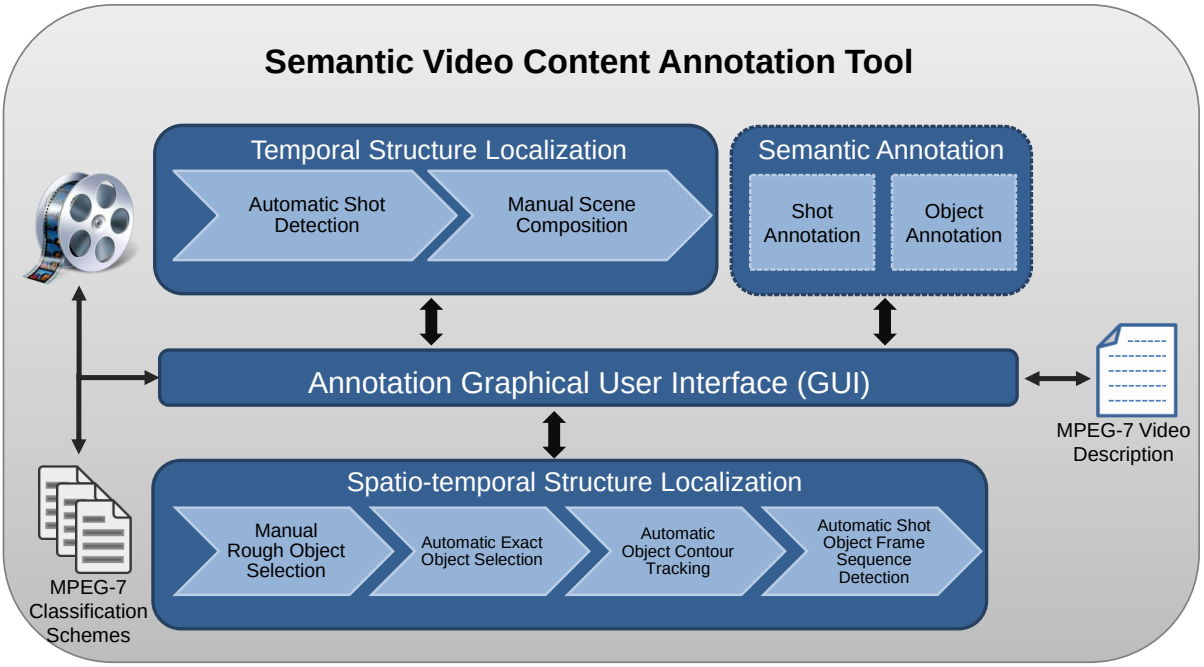


FIGURE 11.2 – Conceptual architecture of SVCAT

the shot on the right side of the button gets the same scene number as the shot on the left hand side, thus grouping them under one scene. A green button means that the shots are connected, otherwise it is red.

Once the scenes are constructed, a temporal structure description in MPEG-7 is generated by clicking the button 'Generate XML'. For instance, Listing 7 shows an excerpt of the generated MPEG-7 description from the example presented in Annex B. Based on this MPEG-7 description metadata, the temporal structure of the video is displayed to the annotator. Figure 11.5 illustrates the scene panel of SVCAT. The top part enables the annotator to navigate through the scenes, while displaying their shot organization, in the bottom part. Moreover, the interface allows the annotator to assign priorities from the priority CS, and free text annotation at the shot-level.

11.3.2 Spatio-temporal Structure Localization and Annotation

11.3.2.1 Object selection approach

As stated in section 11.2.2, two types of approaches are good candidates for SVCAT's object selection function, namely graph cuts and active contours. To select one for implementation in SVCAT, we chose representative algorithms of each approach, implemented them and compared them with respect to segmentation accuracy and performance. For graph cut based approaches, we chose the GrowCut algorithm [137]

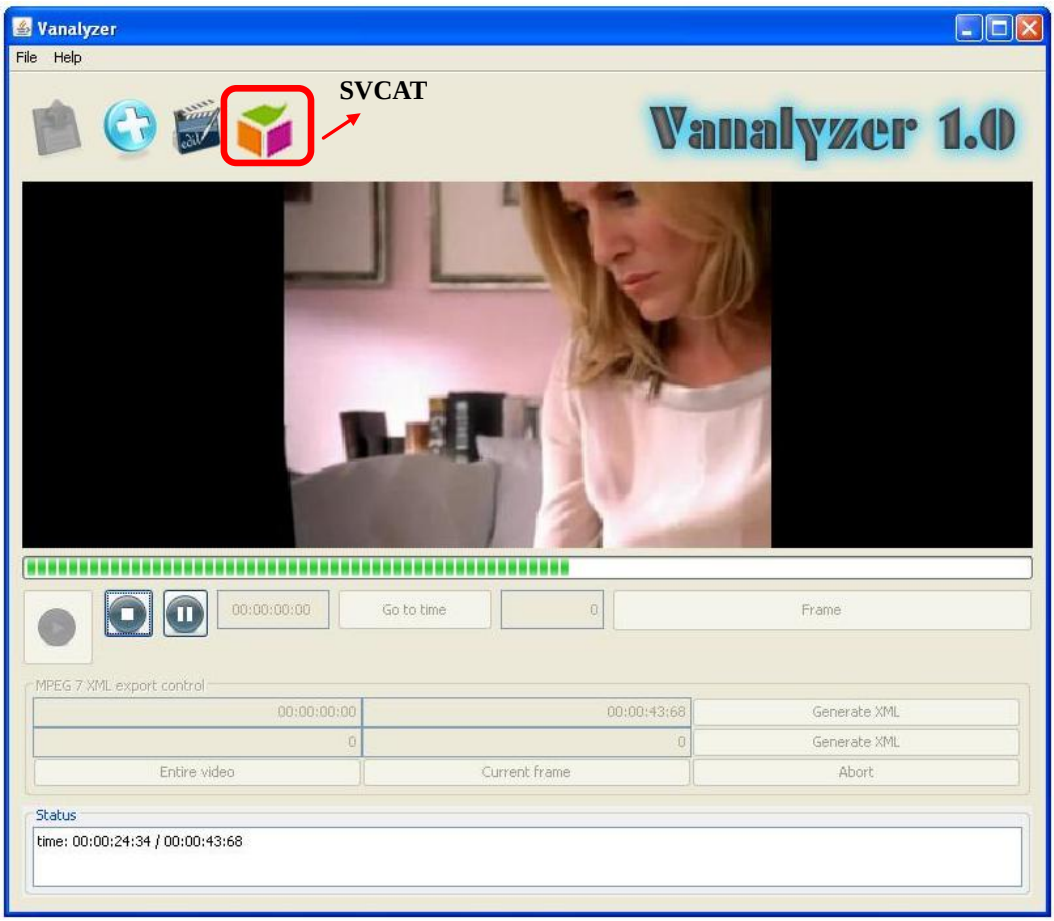


FIGURE 11.3 – Graphical User Interface of VAnalyzer

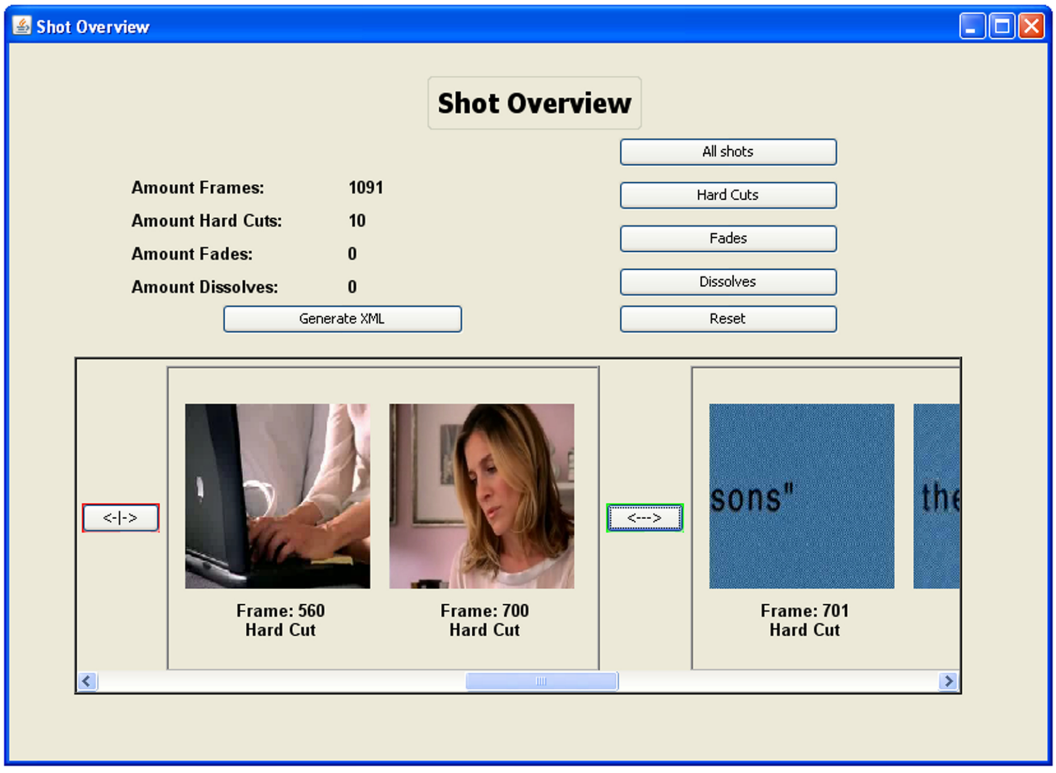


FIGURE 11.4 – Overview of the shots detection in VAnalyzer

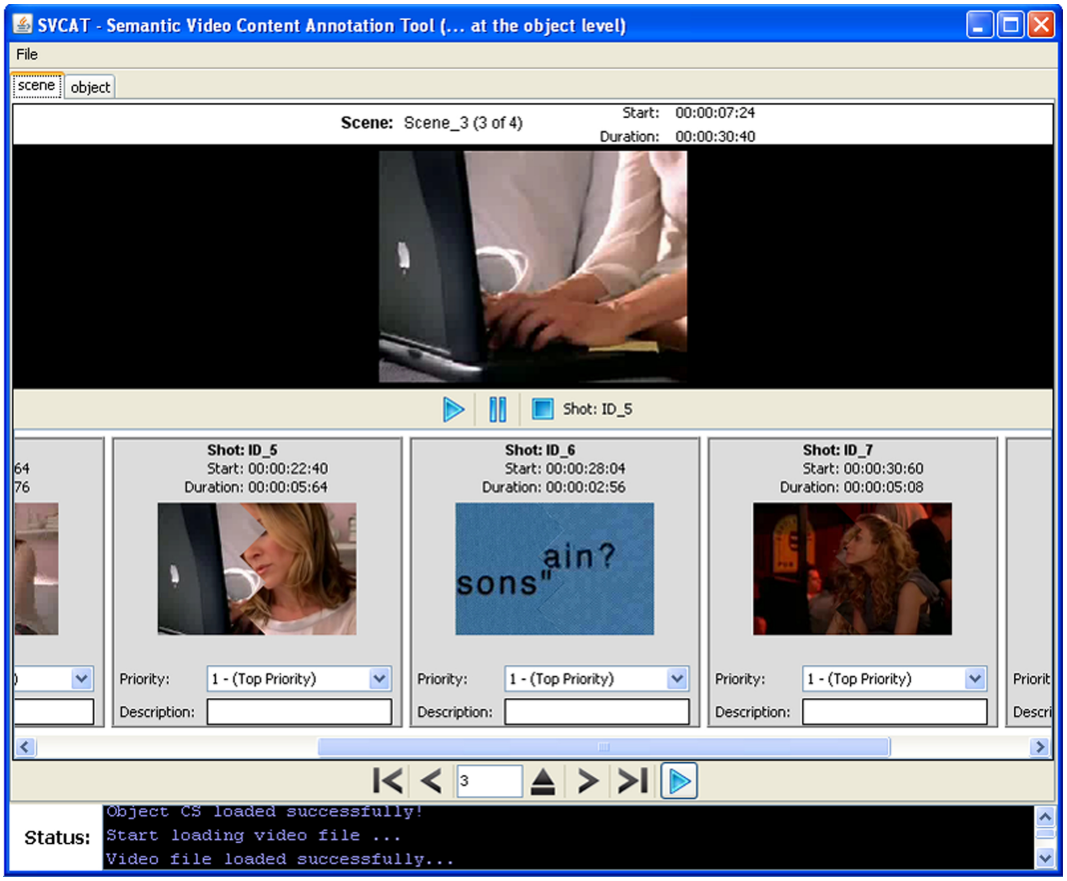


FIGURE 11.5 – Scene panel of SVCAT

and for contour evolution techniques we opted for a level set based snake implementation [72] using the Chan/Vese energy [17], expressed below in equation 11.2.

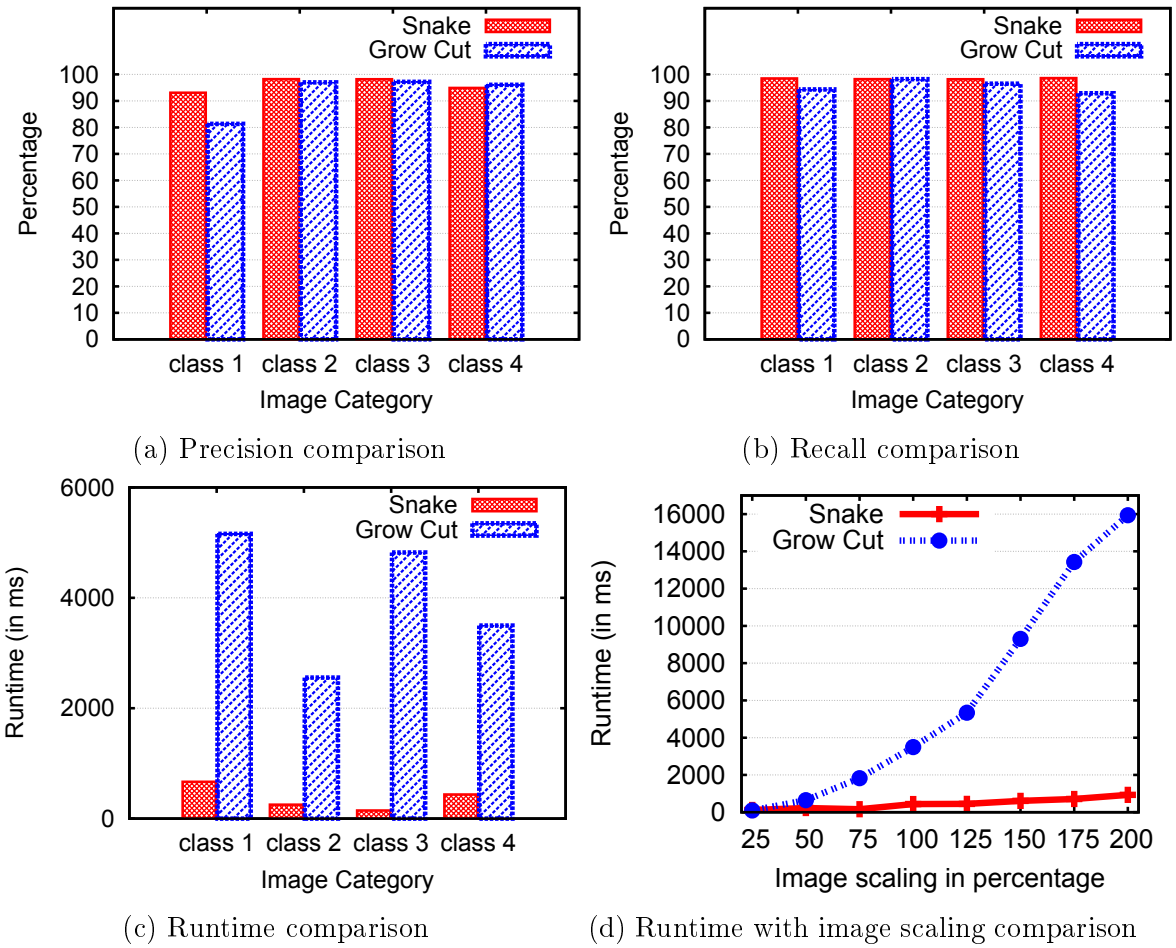
$$E = \int_{interior} (I - \mu_1)^2 + \int_{exterior} (I - \mu_2)^2 \tag{11.2}$$

The experimental set-up for comparing the two approaches consists of four classes of five images each. As depicted in Table 11.2, the classes represent different uniformity combinations (i.e., heterogeneous vs. heterogeneous) with respect to the color and texture characteristics of object and background.

TABLE 11.2 – The experimental classes.

Object/Background	Homogeneous	Heterogeneous
Homogeneous	Class 1	Class 3
Heterogeneous	Class 2	Class 4

To quantitatively evaluate the segmentation accuracy of both approaches, we compare the segmented image against the manually-segmented reference image (often referred to as ground truth), which we represented as binary masks. These masks enable the computation of the precision and the recall measures at pixel-level accuracy. As shown in Figure 11.6 (a-b), the segmentation results of the Snake algorithm are slightly better than the one of GrowCut. With respect to the performance evaluation, we calculate for each class the average of the segmentation time in milliseconds. As illustrated



(a) Precision comparison

(b) Recall comparison

(c) Runtime comparison

(d) Runtime with image scaling comparison

FIGURE 11.6 – Quantitative comparison of the object selection : Snake against GrowCut.

in Figure 11.6 (c), the Snake algorithm outperforms GrowCut. Moreover, we proved that the segmentation time required using Snake is independent from the image size (see Figure 11.6 (d)). Indeed, we evaluate the time using the images of Class 1 with an increasing image scaling from 25% to 200% with step of 25%. The resultant curve can be explained as the snake only performs calculations along the contour line, while GrowCut analyses each pixel within the image. Based on these experimental results, we decided to integrate the Snake approach in SVCAT.

11.3.2.2 Object tracking approach

Regarding the object tracking approach, we have implemented the method proposed by Shi and Karl [114] as we discussed in section 11.2.2. The algorithm assumes that each scene of the video is composed of a background region Ω_0 and an object region Ω_1. The contour of Ω_1 is denoted as C_1. Each of the two regions is modeled with a feature distribution $p(\underline{v} \,|\, \Omega_x)$, where $\underline{v}$ is the feature vector defined at each pixel. In our implementation we used the *hsv* color space and a pixel level texture descriptor [4]. Assuming that the feature distribution in each pixel is independent, the tracking can be regarded as the minimum of the following region competition energy (equation

11.3).

$$E = \underbrace{-\sum_{i=0}^{1} \int_{\Omega_i} \log p\left(\underline{v}\left(\underline{x}\right) \mid \Omega_i\right) dx}_{E_d} + \underbrace{\lambda \sum_{i=0}^{1} \int_{C_i}}_{E_s} \tag{11.3}$$

which results in the following speed functions (equations 11.4 and 11.5),

$$F_d = \log\left[\frac{p\left(\underline{v}\left(\underline{x}\right) \mid \Omega_1\right)}{p\left(\underline{v}\left(\underline{x}\right) \mid \Omega_0\right)}\right] \tag{11.4}$$

$$F_s = \lambda\kappa \tag{11.5}$$

F_d represents the competition between the two regions and F_s smoothly regularizes the contour.

A nice feature of this algorithm, in the context of the integration into SVCAT, is the fact that it also uses level sets to represent the contour. Thus, it is easy to transform the contour output of the Snake algorithm to the representation that is necessary for the tracker.

11.3.2.3 Object Annotation Approach

In order to perform object annotation and tracking, the annotator must choose a scene from the scene panel and switch to the object panel. Figure 11.7 shows the object panel of SVCAT, which implements the object annotation and tracking functionalities. For instance, let us annotate the Apple company logo visible on the back of the laptop, which appears in the frames of the fifth shot within the third scene. Once the required classification scheme are loaded (a), we navigate through the frames of the selected scene to determine the start-frame of the object that we want to annotate (b). Before assigning semantics to an object, spatial segmentation information has to be extracted. To begin, we manually initialize the localization of the spatial object, by using either the rectangles, ellipse or polygon drawing tools (c). Afterwards, we associate semantics to the selected spatial object by choosing a descriptive keyword from the currently loaded classification scheme. Based on this rough selection, we click on the *Snake-Button* that automatically calculates the exact contour selection of the spatial object (c). Once the initial contour has been selected successfully, we track the spatial objects throughout the whole frames of the current scene. Indeed, a click on the *Track-Button* symbolized by a hand in (d), will launch the tracking. The generated spatio-temporal description of this process consists of the size and the contour of the spatial object in each frame in which it appears, as well as description of its associated object frame sequences. These descriptions are represented using the descriptors in the MPEG-7 standard. We refer the reader to Section B.3 for a detail description about the object metadata. Note that the object annotation process can be repeated in order to describe further objects within the video.

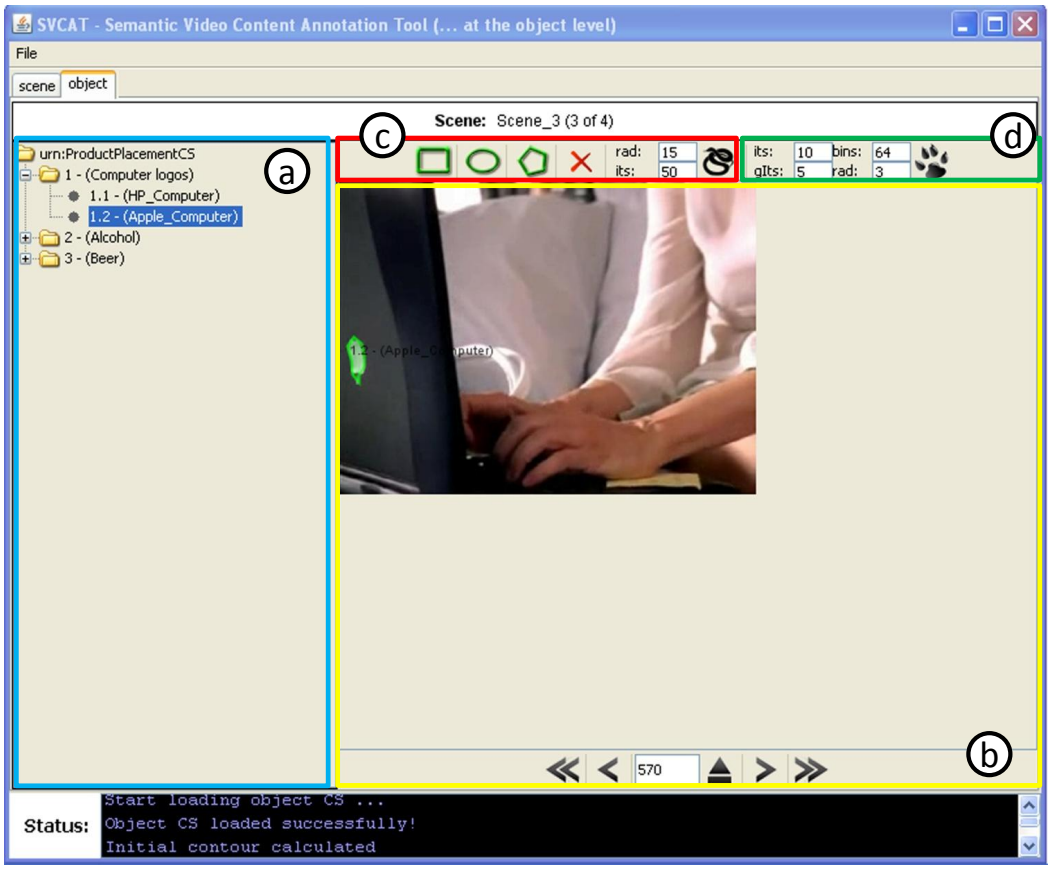


FIGURE 11.7 – Object panel of SVCAT

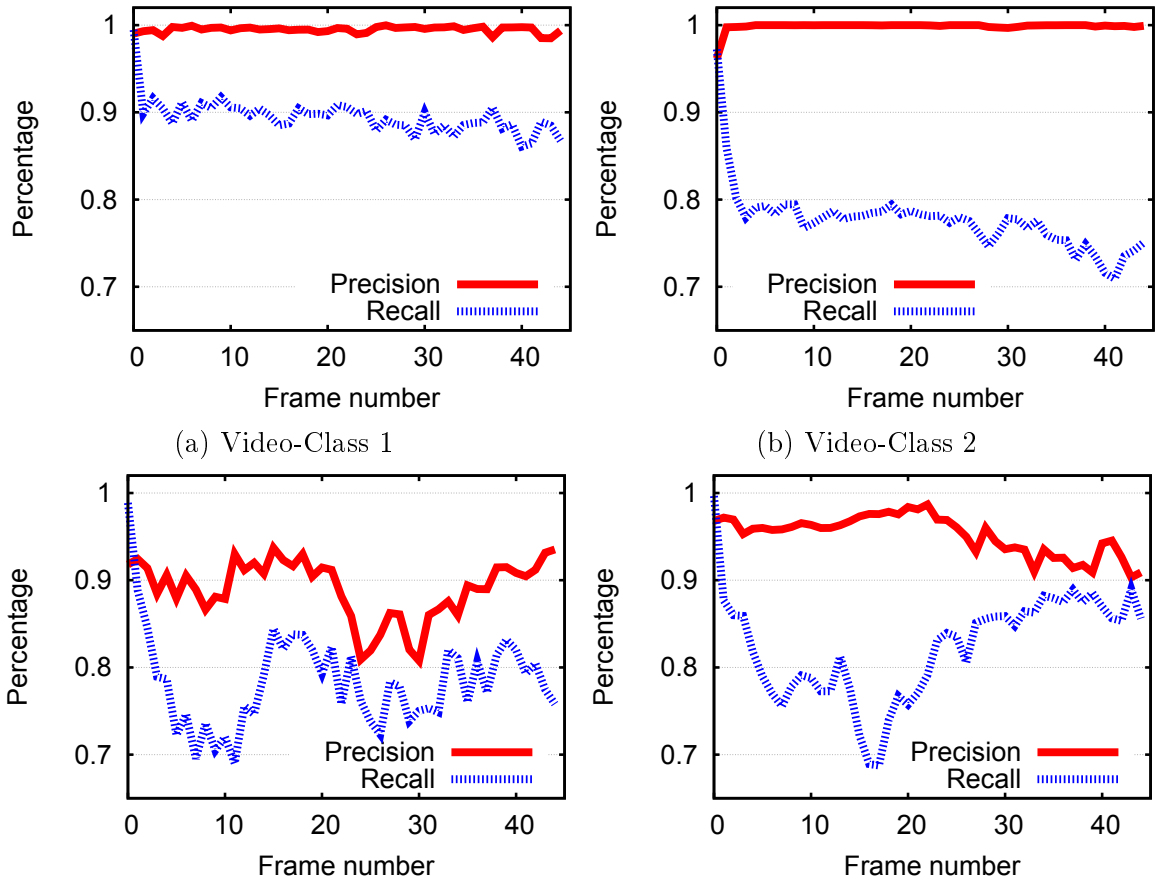
11.4 Experimental evaluation

In this section, we present the results of the evaluation of the accuracy and performance of SVCAT. The experiments were run on a 2.2 GHz dual core machine with 2 GB of RAM. All the algorithms presented in this paper were developed in Java version 1.6 and ran on Windows XP as an operating system. 1 GB of RAM was allocated to the Java Virtual Machine.

The data set comprises four videos, each one representing a different class (see Table 11.2). All videos are in DivX format stored as AVI with a resolution of 320*240 pixels and a frame rate of 25 Fps. For each frame holding the object, we manually segmented it and generated the binary mask of the foreground. As this procedure is time-consuming, we only segmented the object in 45 frames.

To begin with, we studied the accuracy of the contour tracking algorithm for deformable objects taken by a moving camera. As the tracking result of the objects can be stored in a binary mask, we used the same evaluation methodology as for the image segmentation in Section 11.3.2.1. For each frame, we compared the segmentation results of the tracker against manually-segmented reference frame, and computed precision and recall at pixel-level accuracy. Besides accuracy, we also evaluated the runtime performance of the tracking algorithm. For each of the video sequences, we launched three iterations of the contour object tracking and measured the average runtime in milliseconds per frame.

Comparative results of the accuracy evaluation are illustrated in Figure 11.8. It



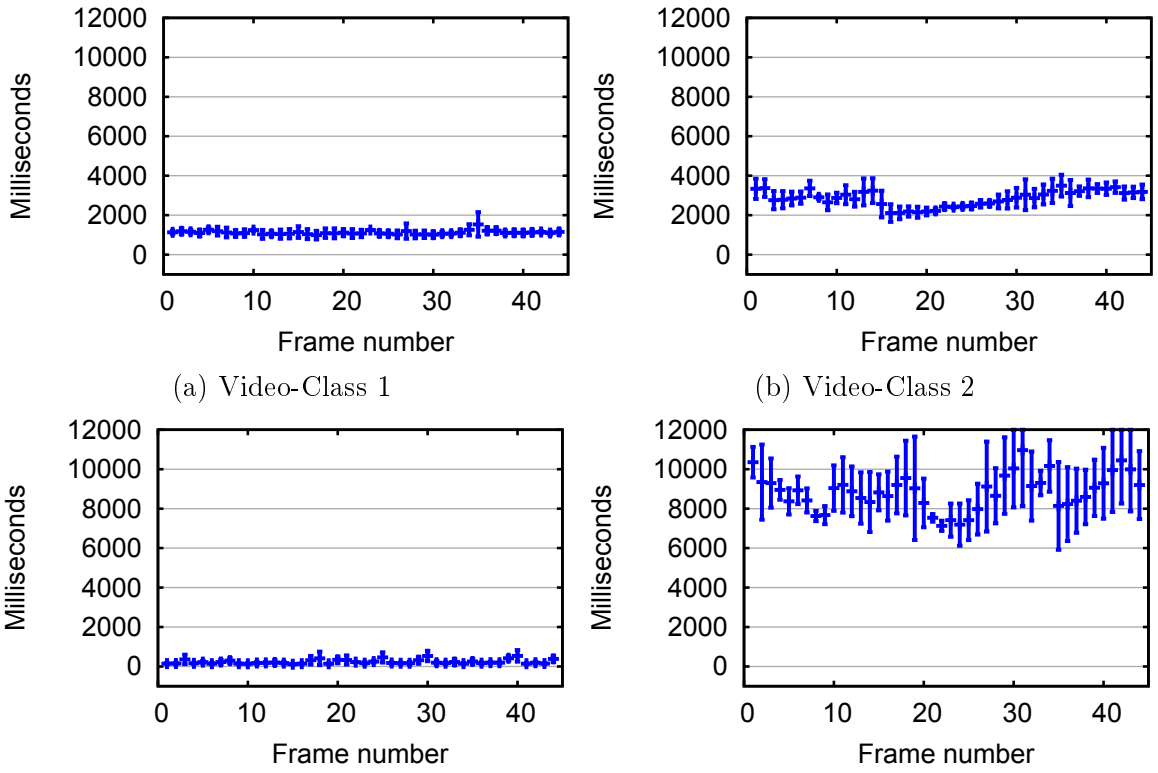
(a) Video-Class 1 (b) Video-Class 2

(c) Video-Class 3 (d) Video-Class 4

FIGURE 11.8 – Accuracy evaluation of the tracking algorithm.

can be observed from the precision-recall curves that the tracking algorithm returned more relevant results than irrelevant, such that not all the relevant pixels are returned. For instance, the precision values reside inside a range of 90% up to 100%, while the recall values reach an average of approximately 80%. This is due to the contour evolution process according to the calculated energy, which is described in Section 11.3.2.2. Indeed, the texture description and the feature representation of a pixel within a particular frame (n) rely on the luminance characteristics of its neighborhood. Thus, the texture descriptor for object pixels in areas close to the contour line might also incorporate background pixels to calculate the feature. This can result in an imprecise description of such pixels yielding in a slightly distorted feature distribution for the object region. Due to this distribution, the contour evolution can sometimes regard object pixels in the consecutive frame $(n+1)$ as background pixels erroneously.

An additional conclusion drawn from these curves is that tracking videos of Class 1 (Figure 11.8 (a)), which consist of homogeneous object and background, obtain better results than tracking videos with heterogeneous regions (Figure 11.8 (b-c-d)). Due to heterogeneity (e.g., different colors, various textures), the feature representation for both object and background regions is not that distinctive as compared to homogeneous conditions (e.g., a single color hue, smooth texture). As a result, the values in the histogram (i.e., the feature distribution) will be scattered across a larger range. As a consequence, the failure rate increases with the contour evolution from one frame to another, since pixels' region membership in the consecutive frame is estimated based on this feature distribution.



(a) Video-Class 1 (b) Video-Class 2

(c) Video-Class 3 (d) Video-Class 4

FIGURE 11.9 – Performance evaluation of the tracking algorithm.

With respect to the runtime evaluation, the results are depicted in Figure 11.9. By examining the performance curves of each video class, we easily observe the tremendous differences between their tracking times, although they have the same resolution. For instance, the average runtime per frame for Video-Class 1, Video-Class 2, Video-Class 3 and Video-Class 4 is about 1.118, 2.857, 0.233 and 8.852 sec, respectively. This can be referred to the calculation of the texture feature during the tracking. In fact, in order to obtain a good tracking accuracy, we used different radii 3, 6, 1 and 7 for Video-Class 1 to Video-Class 4 respectively, in our evaluation. Yet, SVCAT enables the adjustment of the neighborhood radius of pixels that should be relevant for the texture description of a particular pixel. Hence, it became apparent that tracking performance massively depends on the amount of pixels that contribute to the texture description.

Although the tracking runtime performance deteriorates with heterogeneous regions, we argue that the experimental results are quite acceptable for the purpose of PIAF. Indeed, SVCAT aims to automatically provide object localization at pixel-level accuracy in each frame. To achieve such a strong requirement on precision, we consider that relatively long computation times are reasonable.

11.5 Synthesis

In this chapter, we presented the Semantic Video Content Annotation Tool (SVCAT), which was implemented for the purpose of PIAF. SVCAT is a highly automated (i.e., semi-automatic), standard compliant (i.e., MPEG-7) and very accurate (i.e., object level granularity at pixel precision) annotation tool. Based on the video model

presented in Chapter 7, SVCAT makes use of the MPEG-7 descriptions tools and generates standardized annotations at different granularities. Particularly, it achieves a semi-automatic annotation at the object level. To this end, it first performs an automatic exact selection of the object contour based on an initial rough selection given by the user. Then, it automatically propagates object properties to other frames in which it appears using a contour evolution tracking algorithm. By automating these two processes, we have extensively reduced the manual annotation time required. More, SVCAT enables the semantic annotation of objects using a controlled vocabulary held by an MPEG-7 Classification Scheme (CS). Other than the default CS provided with SVCAT, the tool also allows the annotator to explicitly supply his/her own CS. As stated in Chapter 5, the use of CS is highly important since it alleviates the problem of interoperability between the PIAF modules.

Moreover, SVCAT provides functionalities for frame-accurate and key-frame navigation through the temporal structure of the video (i.e., scene, shots) via a user-friendly interface using a time-line. Existing MPEG-7 descriptions can also be imported, enabling a more comfortable gradual annotation process. These functions lighten the task of the annotator in associating the metadata with the video structure and in updating it.

Finally, in order to justify our design choices for each part of the prototype, we have conducted an analytical and experimental evaluation of the existing approaches. Then, we have performed a global evaluation of SVCAT proving the accuracy of its metadata, and that this tool provides a reliable input to PIAF. Indeed, the experimentation results showed that the precision values vary between 90% and 100% according to the texture of the object versus background, while the recall values achieve an average of approximately 80%.

Chapter 12

Context Description and Filtering

In this chapter, we describe the use of MPEG-7 and MPEG-21 description tools to represent the profile of the users including their semantic constraints. In particular, we present an extension of the MPEG standards to include the semantic constraints under a new type of preferences called UsageSemanticPreferences. Moreover, we describe the representation of the information related to the semantic constraint instantiation process that is, the InstantiatedConstraint. The user profile and the semantic constraint instantiation process described in this chapter implement the semantic constraint model presented in Chapter 8.

12.1 User Profile Description

In the following, we describe the use of MPEG-7 and MPEG-21 to represent the profile in PIAF. All the proposed XML schema have been validated against the schemas of MPEG-7 [52] and DIA [16].

12.1.1 General Profile

As stated in Section 3.2 of Chapter 3, MPEG-21 DIA provides the Universal Environment Description (UED) tools to describe the user preferences, terminal capabilities, network conditions and the natural environment. In particular, it defines the *UserType* tool to describe user characteristics, which may be related to general user information, content preferences, etc. Using the constructs provided by the Backus-Naur Form (BNF), the syntax of this tool is given by :

$$\text{User} : := \text{UserCharacteristic}*$$
$$\text{UserCharacteristic} : := \text{UserInfo} \mid \text{UsagePreferences} \mid \text{UsageHistory} \mid \ldots$$

Going back to our application scenario 1 presented in 2.2.1, Listing 12.1 illustrates the profile of the user Pascal using the UED tools. The *UserInfo* (line $7-13$) describes the characteristics of the user 'Pascal' as *GivenName*='Pascal' and *FamilyName*='Doe'. Moreover, Pascal's preferences are defined using the *UserPreferences*

tool. The latter is defined in MPEG-7 as a Description Scheme, enabling users to specify their preferences on ways to consume and browse content.

Listing 12.1 – Illustration of the user profile

```
<?xml version="1.0" encoding="UTF-8"?>
<dia:DIA xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-DIA-NS UED-2nd.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS" xmlns:mpeg7="
    urn:mpeg:mpeg7:schema:2004">
    <dia:Description xsi:type="dia:UsersType">
        <dia:User>
            <!-- Description of the user personal information -->
            <dia:UserCharacteristic xsi:type="dia:UserInfoType">
                <dia:UserInfo xsi:type="mpeg7:PersonType">
                    <mpeg7:Name>
                        <mpeg7:GivenName>Pascal</mpeg7:GivenName>
                        <mpeg7:FamilyName>Doe</mpeg7:FamilyName>
                        ...
                    </mpeg7:Name>
                </dia:UserInfo>
            </dia:UserCharacteristic>

            <!-- Description of the user preferences -->
            <dia:UserCharacteristic xsi:type="dia:UsagePreferencesType">
                <dia:UsagePreferences>
                    <mpeg7:FilteringAndSearchPreferences> ... </
                        mpeg7:FilteringAndSearchPreferences>
                    <mpeg7:BrowsingPreferences> ... </
                        mpeg7:BrowsingPreferences>
                    ...
                     <!--  Extending the UserPreferences DS with
                         UsageSemanticPreferences -->
                    <mpeg7:UsageSemanticPreferences> ... </
                        mpeg7:UsageSemanticPreferences>
                </dia:UsagePreferences>
            </dia:UserCharacteristic>
        </dia:User>
    </dia:Description>
</dia:DIA>
```

However, the *UserType* tool cannot express Pascal's semantic constraint that is, 'watching the movie without visually displayed brand names'. Therefore, we propose to extend the *UserPreferences* DS with a new type of preferences called *UsageSemanticPreferencesType*, which is in turn defined as a set of *SemanticConstraints*. The XML schema of this extension is depicted Listing 12.2.

Listing 12.2 – Illustration of the UsageSemanticPreferences

```
<!-- Definition of UserPreferences DS -->
<complexType name="UserPreferencesType">
    <complexContent>
        <extension base="mpeg7:DSType">
            <sequence>
                <element name="UserIdentifier" type="
                    mpeg7:UserIdentifierType" minOccurs="0"/>
                <element name="FilteringAndSearchPreferences" type="
                    mpeg7:FilteringAndSearchPreferencesType" minOccurs="0"
                     maxOccurs="unbounded"/>
```

```
8                   <element name="BrowsingPreferences" type="
                        mpeg7:BrowsingPreferencesType" minOccurs="0" maxOccurs
                        ="unbounded"/>
9
10                  <!-- Extending the UserPreferencesType defined in MPEG-7
                         with UsageSemanticPreferences -->
11                  <element name="UsageSemanticPreferences" type="
                        mpeg7:UsageSemanticPreferencesType" minOccurs="1"
                        maxOccurs="1"/>
12
13              </sequence>
14              <attribute name="allowAutomaticUpdate" type="
                    mpeg7:userChoiceType" use="optional" default="false"/>
15          </extension>
16      </complexContent>
17  </complexType>
18
19  <!-- Definition of UsageSemanticPreferences DS -->
20  <complexType name="UsageSemanticPreferencesType">
21      <sequence>
22          <element name="SemanticConstraint" type="
                mpeg7:SemanticConstraintType" minOccurs="1" maxOccurs="
                unbounded"/>
23      </sequence>
24  </complexType>
```

12.1.2 (User) Semantic Constraint

A (user) semantic constraint specifies the set of objects that the user desires to remove from the video (refer to Section 8.1.1 of Chapter 8). We defined it as a 3-tuple $SemCt = \langle name_{SemCt}, desc_{SemCt}, T_{SemCt} \rangle$. Listing 12.3 illustrates the XML schema that describes the syntax of the *SemanticConstraint*.

Listing 12.3 – Illustration of the SemanticConstraint

```
1   <!-- Definition of SemanticConstraintType -->
2   <complexType name="SemanticConstraintType">
3       <sequence>
4           <element name="SemanticConstraintName" type="string"/>
5           <element name="Description" type="string" minOccurs="1"/>
6           <element name="KeywordList" type="mpeg7:KeywordListType"
                minOccurs="1"/>
7       </sequence>
8       <attribute name="SmCt_ID" type="string" use="required"/>
9   </complexType>
10
11  <!-- Definition of KeywordListType -->
12  <complexType name="KeywordListType">
13      <sequence>
14          <element name="Keyword" type="mpeg7:ControlledTermUseType"
                minOccurs="1" maxOccurs="unbounded"/>
15      </sequence>
16  </complexType>
```

The *Description* corresponds to the meaning of the semantic constraint. It is mandatory as it is displayed to the user in the User Profile Description Interface, representing

the semantic constraint. The *KeywordList* element specifies the set of keywords related to the semantic constraint. Each Keyword is indicated by a reference to a relevant term in the Classification Scheme (CS) defined in MPEG-7. For instance, referring to our use case, DownVid provides the user with a video content adaptation service from product placements. This is translated by a semantic constraint such as the SemanticConstraintName = 'ProductPlacement-Omitted', Description = 'Remove product placement including logo, text, object and video segments', and the list of Keywords Apple-Computer, Vodka, Budweiser. By using the same classification scheme to describe the constraints and annotate the videos (as described in Chapter 8), we can guarantee a consistent expression of user constraints and annotations, thereby allowing for the automatic instantiation of a given user's semantic constraints over a certain video. We refer the reader to Annex C for a complete description of Pascal profile including his semantic constraint.

12.2 Implementation of the Semantic Constraint Instantiation Process

The semantic constraint instantiation process takes place immediately upon the request of a user for a video (refer to Section 8.2 of Chapter 8). The semantic constraint in the user profile is instantiated over the MPEG-7 content description of the requested video, resulting in an instantiation constraint. In the remainder of this section, we first define the syntax of an instantiated constraint, and then describe the implementation of the instantiation process.

12.2.1 Instantiation Constraint Definition

An *InstantiatedConstraint* contains the information that is required by the content adaptation engine to adapt a given video to a given semantic constraint. This information only consists of the description of the scenes and shots holding the object, such that the associated terms to the object are related to the keywords described in the semantic constraint. Listing 12.4 depicts the general structure of the *InstantiatedConstraint*.

As can be observed from the listing, the root element *InstantiatedConstraint* contains information about the scenes of the video referenced by Video_REF, which are bound to the *SemanticConstraint* referenced by SmCt_REF. A Scene is described by the position of its start frame, length, and the set of shots affected by the constraint. A Shot is described by the position of its first frame in the video, its length, priority value and the Shot Object Frame Sequence (SOFS). An SOFS element is identified by SOFS_ID and corresponds to one single object matching the constraint in the shot. The SOFS is composed of one or more object frame sequence (ofs) described by a start position, length, and information about its object frames (refer to Section 7.3.2). Besides the position number of the object frame in the video, each one contains information about the localization of the object region, described by the Contour as well as its Size (in pixels). We refer the reader to Annex D, which provides the syntax of the *InstantiatedConstraint* schema.

Listing 12.4 – General Structure of an *InstantiatedConstraint*

```
<InstantiatedConstraint InstCt_ID="InstantiatedConstraintID" SmCt_REF="
    Reference to SemanticConstraintID" Video_REF="Reference to VideoID" >
    <Scene Scene_ID="SceneID" Scene_SFrame="Starting Frame of Scene"
        Scene_Length="Length in Frame of Scene" >
        <Shot Shot_ID="ShotID" Shot_SFrame="Starting Frame of Shot"
            Shot_Length="Length in Frame of Shot">
            <SOFS SOFS_ID="SOFSID">
                <ofs ofs_ID="ofsID" ofs_SFrame="Starting Frame of ofs"
                    ofs_Length="Length in Frame of ofs">
                    <ObjFrame ObjFrame_NB="18" >
                        <ObjectSize> Size of the object in the frame </
                            ObjectSize>
                        <ObjectPriority> Priority of the object in the
                            frame </ObjectPriority>
                        <ObjectContour> Description of the contour of the
                            object in the frame </ObjectContour>
                    </ObjFrame>
                    <ObjFrame ObjFrame_NB="19" > ...
                </ofs>
                <ofs ...
            </SOFS>
        </Shot>
        <Shot ...
    </Scene
    ...
    <Scene ...
</InstantiatedConstraint>
```

12.2.2 Constraint Instantiation Workflow

Going back to our application scenario 1 presented in 2.2.1, we describe in the following the workflow of the constraint instantiation process. Following the request of Pascal to watch 'Dark Tree', DownVid retrieves both the UED of Pascal and the MPEG-7 descriptions file of the requested video. The workflow of the instantiation process is illustrated in Figure 12.1. We implemented it using the Java Architecture for XML Binding (JAXB)[12], which can convert Java object to or from XML file. Indeed, we use JAXB to convert (i.e., unmarshal) the XML documents (UED and MPEG-7 video descriptions) into Java objects in order to query them. Moreover, we use JAXB to convert (i.e., marshal) the Java objects into XML using the *InstantiatedConstraint* schema. The successive steps of the instantiation process are described as follows :

- **(1)** The Constraint Instantiation Module queries the UED to extract the keywords of the semantic constraint.

- **(2)** The content description of the requested video is parsed to identify the object with terms that match the keywords of the semantic constraint. This is an unambiguous process as the terms of the object and of the constraint are derived from the same Classification Scheme (CS). The CS also makes it possible to match a general constraint with a more specific object description ; for example, a constraint on Product Placement will match an object described as Apple-Computer, if Apple-Computer is a sub-term of Product Placement in the CS.

12. https://jaxb.java.net/

– **(3)** For each matching of term, the video description is filtered to extract the hierarchical video elements corresponding to the object (scenes, shots, SOFSs, ofss, object frames) as well as all the required information, as described in the previous Section 12.2.1. The result is stored in an XML file following the syntax of the *InstantiatedConstraint* schema.

– **(4)** Finally, the *InstantiatedConstraint* is sent to the Content Adaptation Module to help the ADTE in choosing the adaptation plan.

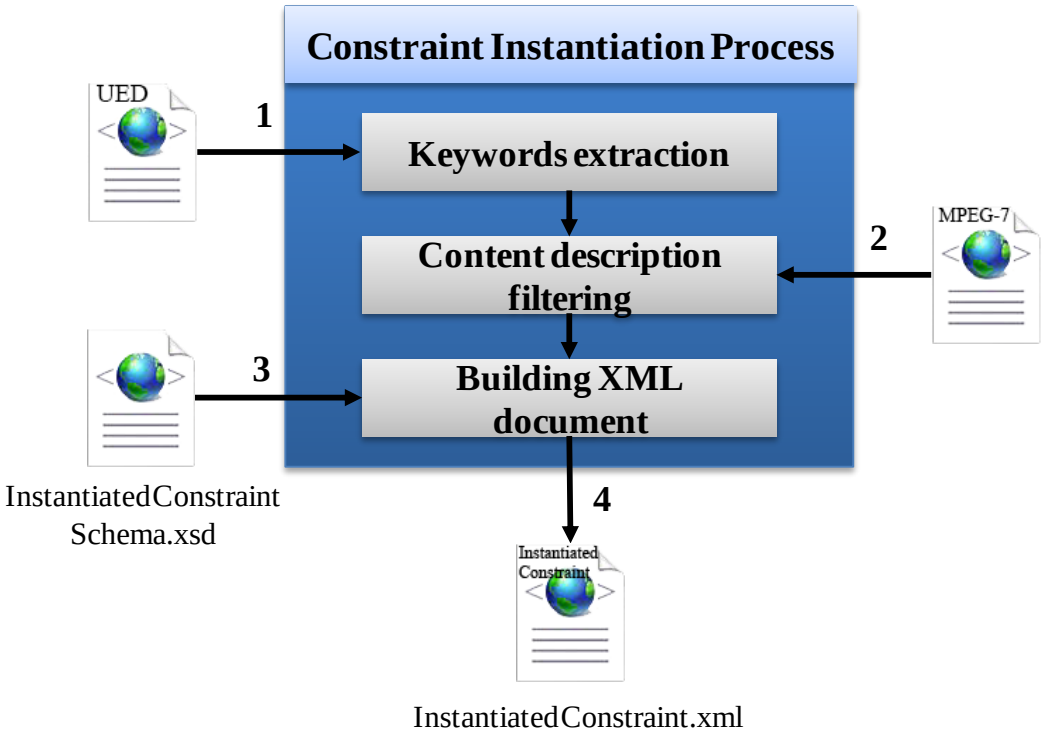


FIGURE 12.1 – Constraint instantiation process.

We note that for the purpose of this thesis, we deal with profile consisting of one semantic constraint that is related to one single object. It is possible to deal with a profile containing several semantic constraints, only if these constraints require the adaptation of shots that do not overlap. In this case, the instantiation process is repeated for each semantic constraint and the results are merged in one InstantiatedConstraint.

12.3 Synthesis

In this chapter, we presented the implementation of the user profile and its associated semantic constraints, compliant to the model presented in Chapter 8. For the purpose of PIAF and in order to be compliant to the MPEG standards, an extension to the MPEG-7 and MPEG-21 standards was also proposed to enable the representation of users semantic constraints. Furthermore, we described the implementation of the instantiation process by the use of JAXB, as well as the syntax of the *InstantiatedConstraint*.

Part IV

Evaluation of the Adaptation Decision Taking Engine

Chapter 13

Utility Function Evaluation by Subjective Testing

This chapter describes the subjective assessment method that we used to evaluate the performance of the Utility Function (UF) model. The performance was determined from a comparison between viewer ratings obtained in controlled subjective tests and quality predictions from the UF model. This analysis demonstrates that the UF model is a good predictor of the QoE reported by the users. The predicted ratings show a strong linear correlation of 0.84 *with the Mean Opinion Score (MOS) ratings. Moreover, we show that the linear model for the utility function performs very well compared to* 31 *different polynomial models of degree up to* 2*, and sometimes it performs the best. Finally, we show through the evaluation of different video content adaptation methods that spatio-temporal adaptation methods outperform those in which only a single adaptation dimension (spatial or temporal) is used.*

13.1 Introduction

The purpose of conducting subjective video quality evaluation of different video content adaptation methods is threefold :

1. outlining the importance of having a utility function model that considers the correlations of affected area, affected priority area, spatial perceived quality, temporal perceived quality and processing cost, in order to decide on the best adaptation plan.

2. evaluating the accuracy of the utility function model in predicting the adaptation plan that maximizes the user-perceived quality.

3. comparing the video quality of a spatio-temporal adaptation with conventional adaptation strategies in which only a single adaptation dimension (spatial or temporal) is used to adapt the video.

13.2 Test Methodology

This section describes the test conditions and procedures used in this test to validate and evaluate the performance of the utility function and its parameters proposed in this thesis.

13.2.1 Test Video Preparation

Three video sequences $V1$, $V2$ and $V3$ extracted respectively from the films 'Casino Royale and Quantum of Solace', 'Transformers' and 'Sex and the City' were chosen for this experiment as they contain a lot of products placement. The duration of a video, the total number of its shots as well as the number of shots including advertisements, vary from one to another (see Table 13.1). Since the duration of the video sequences is short, we assume that each video consists of one single scene. Therefore, the adaptation computation is applied on the video. Moreover, these videos were carefully prepared so that they contain a wide range of spatial and temporal complexity. Indeed, the aim of this evaluation is to outline the importance of the parameters in the UF model. For instance, some of the shots to be adapted in $V1$ were chosen to be prior to the meaning of the video story, and some had fast movement where the salient object disappears in the middle of the shot. In $V2$, the shots were chosen such that the salient object is of a small size and appears in nearly all the shots. $V3$ combines the complexity of both $V1$ and $V2$, and it additionally contains objects of a large size, such that the artifacts caused by the inpainting algorithm are perceivable.

TABLE 13.1 – Videos used in the evaluation.

	Duration	**Total number of shots**	**Total number of shots to adapt**
V1	1 min 09 sec	26	5
V2	58 sec	14	6
V3	1 min 34 sec	15	9

13.2.2 Observers

As stated in the ITU-T Recommendation P. 910, 2008 [60], the number of observers in a specific test should really depend on the required validity and the need to generalize from a sample to a larger population. It is recommended that at least 15 observers should participate in the experiment. They should not be directly involved in video quality evaluation as part of their work and should not be experienced assessors. For the purpose of this experiment, a total of 21 participants (7 females and 14 males), who are not experienced assessors, were tested. Their ages ranged from 24 to 33, with an average of 28. Only one observer was detected as an outlier and thus eliminated. Indeed, she assesses the first two videos and gives a score of 1 for all the questions related to the third video.

13.2.3 Rating Process

The test consisted of assessing the video adaptation quality result of six different adaptation methods M_j, $j \in \{1,\ 2,\ \ldots,\ 6\}$ on three different videos V_i, $i \in \{1,\ 2,\ 3\}$ with regard to satisfying the following semantic restriction : "Eliminate advertisement from the video content". As shown in Table 13.2, these adaptation methods are classified in two categories : One-dimensional and two-dimensional adaptation technique. For instance, M_1 results in an adaptation plan were only the $Drop - Shot$ operator is assigned to the shots undergoing the adaptation process, while M_4 results in an adaptation plan were the $Drop - Object$ operator is forbidden to be used (refer to Section 10.1). It is worthy to note that *M6* results in an adaptation plan, which we believe it should be generated by our ADTE. We argue that at this stage of the work, it was not possible to calculate the UF and generate the adaptation plan, since some constants of the perceived quality parameters p_3 and p_4, should be first defined experimentally.

TABLE 13.2 – Classification of the adaptation methods.

Adaptation Operations	**One-dimensional adaptation method**			**Two-dimensional adaptation method**		
	M1	*M2*	*M3*	*M4*	*M5*	*M6*
Drop-Shot	x			x		x
Drop-SOFS			x	x	x	x
Drop-Object		x			x	x

For each video evaluation, a playlist of seven videos was shown : the first video was the reference, and the other six videos were the adapted ones. We recall that the subjective evaluation tests were done on pre-adapted videos, instead of doing the adaptation online, because the AEE was not implemented. As each video was played out, the observers were asked to rate the adaptation methods with respect to five subjective metrics SM_k, $k \in \{1,\ 2,\ \ldots,\ 5\}$ listed respectively as follows :

1. Semantic Perceived Quality (SemPQ) ; the aim of rating the SemPQ is to assess whether the adaptation method preserves the semantic integrity of the original video. By this we mean, to rate if much important information was skipped from the video after adaptation.

2. Temporal Perceived Quality (TPQ) ; the observers were asked to express their satisfaction by rating how much the adaptation method delivers a video that fits their perception without noticing gaps (temporal discontinuities) in parts of the video.

3. Spatial Perceived Quality (SPQ) ; the observers were asked to express their satisfaction by rating whether the adaptation method delivers them a video containing visual artifacts like blurring, which are noticeable and are disturbing them.

4. Time Efficiency (TE) ; in order to evaluate the adaptation process performance regarding the waiting time, for each adaptation method, we have computed beforehand the time needed to adapt each video. We translated this time by a

sequence of black frames that we introduced at the beginning of each adapted video. The observers were asked to watch the video from the beginning without forwarding it.

5. Satisfaction of the Overall Adapted Quality (SQ); at the end of each evaluation, the observers were asked to express their satisfaction of the overall adapted content, while considering the aforementioned metrics. The purpose of this rating is to outline the importance of considering the correlation of the five parameters in the UF.

A five-level quality scale ranging from 1 to 5 was used to rate these properties, such that $1 = bad$, $2 = poor$, $3 = fair$, $4 = good$ and $5 = excellent$. For instance, a score of 5 for the SemPQ signifies that the semantic integrity of the adapted video is strongly preserved as it was not adapted at all, while a score of 1 means the semantic integrity was totally destroyed with respect to the reference video. To prevent any ambiguity, an explanation of the rating scores for each property was given. All the questions were mandatory and for each one only one answer should be marked. Then, at the end of the questionnaire, the observers had the freedom to make suggestions and to give their opinion and feedbacks. As reported in their comments, the explanation of the rating scale was very helpful.

13.2.4 Setup of the Testbed

Table 13.3 depicts the setup of the testbed. Indeed, each 3-tuple $\langle V_i,\ M_j,\ SM_k \rangle$ got a rating score $RS_{i,j,k,l}$ from observer l on the subjective metric SM_k for the video V_i adapted by the method M_j. We gathered a total of 1800 rating scores values.

TABLE 13.3 – Summary of the test.

Number of	**value**
videos	3
adaptation methods	6
tested videos	$6 \times 3 = 18$
rated metrics	5
rating scores per metric for a tested video	20
rating scores per metric for whole tested videos	$18 \times 20 = 360$
total rating scores	$360 \times 5 = 1800$

13.3 Data Analysis

This section describes the evaluation metrics and procedure used to assess the performance of the utility function and of its parameters model, as an estimator of the users' ratings scores.

13.3.1 Scaling Rating Scores

As stated in Section 9.2, the scale measurement of the utility function and its parameters is $[0, 1]$. To this end, the rating scores assigned by the observers are re-scaled by linearly mapping the range $[1, 5]$ to $[0, 1]$ applying the following :

$$RS_{i,j,k,l} = \frac{(RS_{i,j,k,l} - 1)}{4} \tag{13.1}$$

13.3.2 Calculating the Mean Opinion Score Values

Recommendation ITU-R BT.500-13 [61] and ITU-T Recommendation P. 910, 2008 [60], provide methodologies for subjective video quality assessment methods for multimedia applications including general methods of test, the grading scales and the viewing conditions. Since our aim is to subjectively evaluate the quality of different adaptation methods of videos, thus we adopt theses methodologies. As suggested in the Recommendations, we calculate the Mean Opinion Score (MOS) of each 3-tuple $\langle V_i,\ M_j,\ SM_k \rangle$ as follows :

$$MOS_{i,j,k} = \frac{1}{N} \times \sum_{l=1}^{N} RS_{i,j,k,l} \tag{13.2}$$

Where N is the number of observers, that is 20 in our case, and $RS_{i,j,k,l}$ is the rating score of observer l of the subjective metric SM_k for the video V_i adapted by the method M_j.

Furthermore, based on [61], we calculate for each $MOS_{i,j,k}$ its associated Confidence Interval (CI) denoted by $CI_{i,j,k}$, which is derived from the Standard Deviation (STD) and size of each sample N. We propose to use the 90% confidence interval that is given by :

$$[MOS_{i,j,k} - CI_{i,j,k},\ MOS_{i,j,k} + CI_{i,j,k}]$$

$$\begin{aligned} &where && CI_{i,j,k} = 1.729 \times \frac{STD_{i,j,k}}{\sqrt{N}} \\ &and && STD_{i,j,k} = \sqrt{\sum_{i=1}^{N} \frac{(MOS_{i,j,k} - RS_{i,j,k,l})^2}{(N-1)}} \end{aligned} \tag{13.3}$$

such that $STD_{i,j,k}$ is the standard deviation for each 3-tuple $\langle V_i,\ M_j,\ SM_k \rangle$. In addition, since we are dealing with a small sample $N = 20$, we use the Student's t-distribution, where the value 1.729 is the selected value for t-distributions with the number of degrees of freedom equals to $N - 1 = 19$ for a range of 90% two-sided. Then, the absolute value of the difference between the experimental mean score and the 'true' mean score (for a very high number of observers) is smaller than the 90% confidence interval.

13.3.3 Evaluation Metrics

In order to evaluate the performance of the UF and its parameters, we follow the methodology suggested by the video quality experts group (VQEG) [44]. Indeed, we evaluate the ability of our models to estimate a value, that is similar to the MOS values given by the observers, with respect to three aspects :

a. Prediction accuracy : it is the ability of a model to predict the subjective quality ratings MOS with low error. The prediction accuracy of a given model depends on how strong the linear relationship is between the MOS and the values predicted by a model. The strength of this relationship is measured by the correlation coefficient. Several different correlation coefficients can be calculated for scales of measurement for the variables on X-and Y-axes. In order to choose the appropriate one, we relied on the matrix defined in [49]. Since the scale of measurement for both MOS and the values of our proposed models is always an interval $[0, 1]$, we chose to calculate the Pearson linear correlation coefficient (r). The interpretation of the size of a correlation coefficient was based on the rule of thumb described in [49]. Indeed, its size varies between -1 and 1 such that a higher positive or negative value means a stronger positive or negative correlation. Moreover, as we are dealing with a small sample size, we need to prove that the correlation is statistically significant. Thus, to assess the statistical significance of our correlation, we calculate the $p - value$ alongside with the Pearson coefficient. Both values $p - value$ and r are computed by performing the t-test in Excel. According to the interpretation description in [49], a lower $p - value$ (for our case less than 0.05) means that the correlation is statistically significant, so we can use the calculated Pearson coefficient.

Although the Pearson correlation coefficient indicates the strength of a linear relationships between two variables, its value can be misleading depending on the distribution of the data. For instance, Figure 13.1 illustrates four different data sets of (x, y) pairs created by Francis Anscombe [8]. Each of these data sets yields the same standard output from a typical regression program, namely the mean of both the x's and the y's, the variance, the correlation, the equation of regression line : $y = 3+0.5x$, the estimated standard error, etc. Francis Anscombe underline the fact that both calculations and graphs should be studied. Indeed, as can be observed from the figure [13], only the first plot (top left) corresponds to what one would expect from linear relationship. However, in the other plots, Francis Anscombe shows that one outlier is enough to produce a high correlation coefficient, even though the relationship between the two variables is not linear.

Besides the interpretation of both calculations and graphs, it is also important to study the monotonic relationship between the values predicted by a model and the one given by the observers. Indeed, having a moderate value of Pearson coefficient (i.e., $\pm.50$ to $\pm.70$), we can only conclude that the variables are not correlated in a linear fashion. However, these variables can be consistently correlated but in another fashion. For subjective video quality assessment methods, this information is valuable since it indicates that the model should be reconsidered to accurately predict the MOS values.

b. Prediction monotonicity : this is the degree to which the predictions of a model

13. This figure was taken from Wikipedia http://en.wikipedia.org/wiki/File:Anscombe%27s_quartet_3.svg

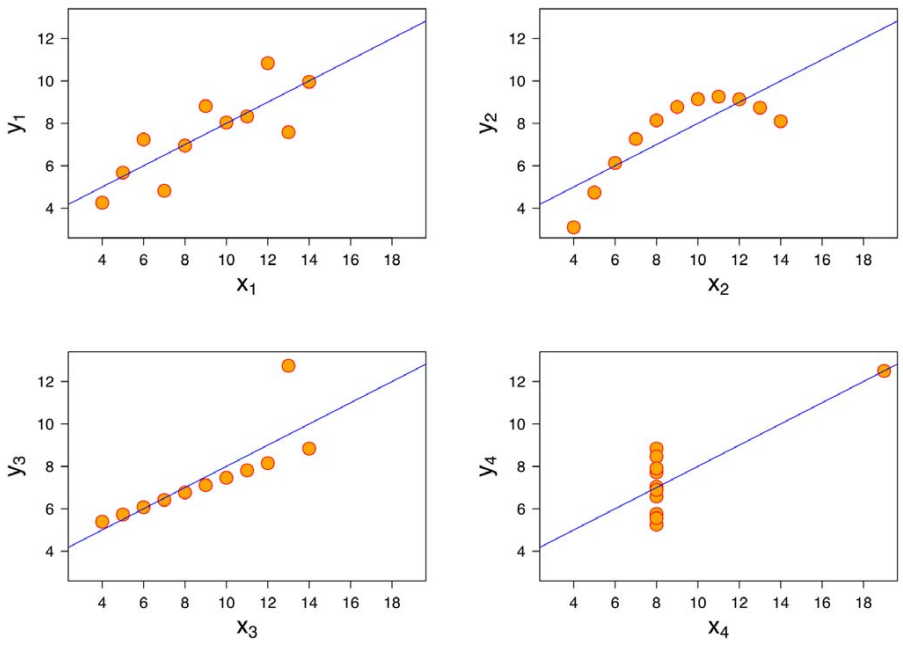


FIGURE 13.1 – Four sets of data with the same correlation of 0.816 as described in [8].

agree with the relative magnitudes of subjective quality ratings MOS. In simpler words, it is to measure the strength of a monotonic relationship between the values predicted by a model and the one given by the observers. Indeed, for subjective video quality assessment methods, it is important to know if when one MOS value increases, so does the predicted one, or vice-versa. Spearman rank-order correlation coefficient(r_s) between the MOS and a model values, is generally responsible for providing an evaluation of the prediction monotonicity. A positive (resp. negative) Spearman correlation coefficient corresponds to an increasing (resp. decreasing) monotonic trend between the two data set. Similar to the Pearson coefficient, the interpretation of Spearman coefficient value is also based on the rule of thumb [49]. If both coefficient are very similar and different from zero, there is indication of a linear relationship.

Although the value of the Pearson coefficient can be sometimes misleading, it is more sensitive to outliers than the one of Spearman coefficient. In fact, this latter measures the rank order of the points, and does not care exactly where they are. Thus, we can still obtain a valid result even if we have outliers in our data. Therefore, another aspect to be considered is the evaluation of the errors that yield from the prediction of a model. In simpler words, we need to measure how away the predicted values are far from the opinion score given by the subject themselves.

c. Errors Evaluation : the error of a model is measured by calculating the differences between values predicted by the model and the values actually observed. A large difference means a large error, and thus the performance of the model is low. In our case, since the objective models will not be able to predict the average opinion score more accurately than the average subjects themselves, we calculate the Root Mean Square Error (RMSE) to evaluate the errors of our model. Indeed, RMSE is a good measure for the prediction accuracy of a model and is the most important criterion for fit, especially if the main purpose of a model is prediction. Moreover, according to the definition of RMSE in [49], RMSE is negatively orientated, i.e., a smaller value indicates smaller errors between the MOS and the model values.

13.4 Analysis of the Experimental Results

In the remainder of this section, we evaluate the performance of the temporal perceived quality model p_3 after experimentally estimating the decay constant value b of p_3. Afterwards, we describe the process of estimating the values of the weights w_i of our utility function as well as the value of the decay constant value b' of p_4. Moreover, we evaluate the performance of the linearity of the UF model and show that the linear model for the utility function is a good one among other models.

13.4.1 Temporal Perceived Quality

In Section 9.2.4.1, we have argued the choice of the exponential decay model for the temporal perceived quality parameter p_3. In this section, based on the users' rating scores of the adaptation methods with respect to the TPQ metric, we first explain the curve fitting process for finding the decay constant value b of p_3. Finally, we evaluate and discuss the performance of the proposed parameter p_3 model.

13.4.1.1 Estimation of the Decay Constant b

The idea of the curve fitting is to find the value for the decay constant b such that the function p_3 matches the users' rating scores as much as possible. As stated in Section 9.2.4.1, the function p_3 is an exponential decay whose independent variable is $GapSR$ over the domain $[0, 1]$. To this end, for each 3-tuple $\langle V_i,\ M_j,\ SM_2\rangle$ such that $i \in \{1,\ 2,\ 3\}$, $j \in \{1,\ 3,\ 4,\ 5,\ 6\}$ and $SM_2 = TPQ$, we calculate the ratio of the gaps size $GapSR$ in V_i produced by M_j. It is important to note that method M_2 is omitted, since it is related to the $Drop-Object$ operation and does not cause temporal impairments. Moreover, the $GapSR$ is calculated in a video and not in a shot, since the observers rates the TPQ per video and not per shot. Furthermore, for each $\langle V_i,\ M_j,\ SM_2\rangle$, the MOS value of the re-scaled rating scores and its associated Confidence Interval (CI) are calculated according to Equations 13.2 and 13.3.

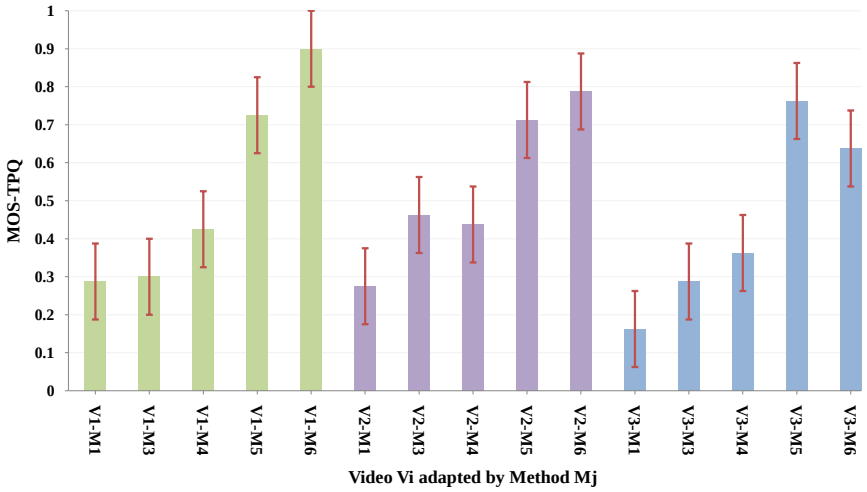


FIGURE 13.2 – The MOS values of the TPQ for 15 adapted videos.

Figure 13.2 illustrates the obtained MOS value of each video V_i adapted by the method M_j. The bar indicates the MOS value, and the red error bar uses the CI to indicate the range which is likely to contain the true mean of the observers' rating scores.

In order to obtain b, the MOS values and the exponential decay function of $GapSR$ data are fitted by the equation below :

$$MOS = exp(-b \times GapSR) \tag{13.4}$$

where MOS is the dependent variable plotted in the y-axis and $GapSR$ is the independent variable plotted in x-axis. The curve fitting was done by the solver tool in Excel while minimizing the sum of squared differences between the actual MOS values and the values predicted by the exponential decay function.

The curve fitting results in a value of 2.8712 for the constant b. In order to evaluate the error during the fitting process, we calculate the Root Mean Square Error (RMSE). Its value is revealed to be very low $\simeq 0,0923$, thus indicating less prediction errors. In addition, we compute the coefficient of determination, denoted R^2 and pronounced R-squared. In statistics, the R-squared is commonly used as an indicator of the goodness of the fit, in the context of statistical models whose main purpose is the prediction. The value of R^2 is revealed to be very high $\simeq 0,8885$, thus indicating a good performance of the exponential decay model. Figure 13.3 illustrates the result of the curve fitting process. The green triangle indicates the value resulting from the fitted exponential decay model, the blue square indicates the MOS value, and the red error bar uses the CI to indicate the range which is likely to contain the true mean of the observers' rating scores.

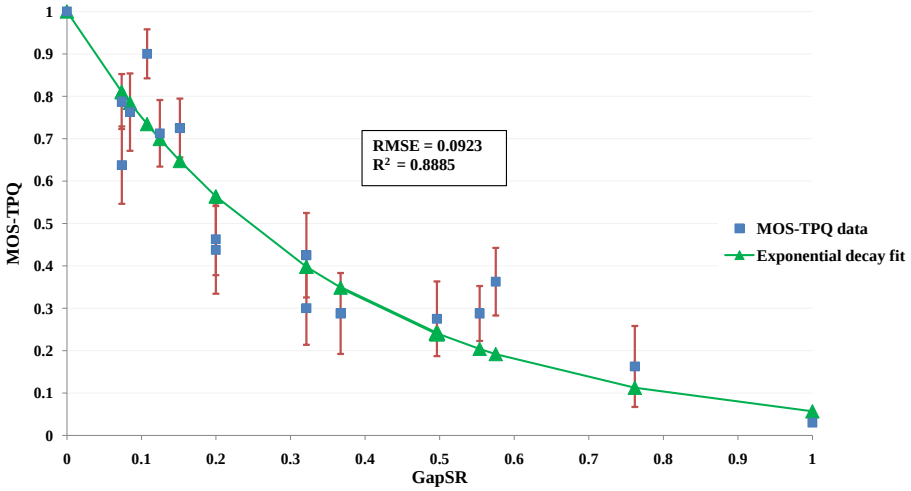


FIGURE 13.3 – Curve fitting for the MOS values and the exponential decay function of $GapSR$ data with $b = 2.8712$.

13.4.1.2 Performance Evaluation of the Temporal Perceived Quality Model

The performance of our temporal perceived quality model is evaluated by depicting the relationship between the values obtained by $p_3 = exp(-2.8712 \times GapSR)$ and the

actual MOS values. In total, we have 15 values corresponding to the 15 adapted videos. As stated in Section 13.4, we evaluate the ability of p_3 to estimate the MOS values with respect to the prediction accuracy, prediction monotonicity and errors evaluation (see Figure 13.4).

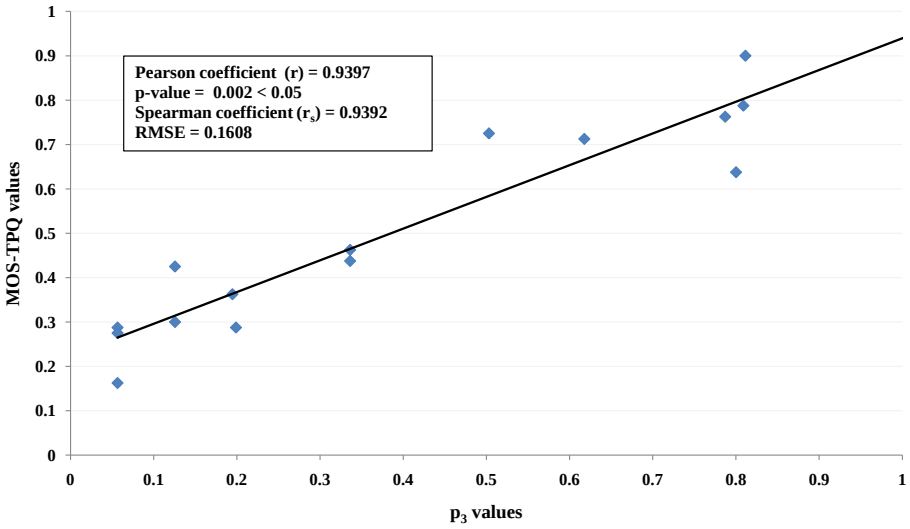


FIGURE 13.4 – Performance evaluation of p_3.

To begin, we perform the t-test in Excel between MOS and p_3 values. The size of the Pearson linear correlation coefficient (r) is found to be $r = 0.9397$ with a $p-value$ equals to 0.002 . These results imply that the correlation is statistically significant (as $p - value = 0.002 < 0.05$), and that p_3 highly correlates with the Mean Opinion Score (MOS) ratings of the human observers (as $r = 0.9397$).

Then, we calculate the Spearman rank-order correlation coefficient(r_s) between the MOS and p_3 values, which is responsible for providing an evaluation of the prediction monotonicity. The calculated value of $r_s = 0.9392$ shows a strong positive correlation.

Finally, we calculate the RMSE to evaluate the errors of our p_3 model. The value of RMSE is found to be 0.1608, which indicates low prediction errors between the MOS and p_3 values, and thus a good performance of p_3 model.

13.4.2 Utility Function

In this section, we aim to evaluate the performance of the linearity of the UF model and to show that the linear model for the utility function is a good one among other models. Based on the users' rating scores of the adaptation methods with respect to the Satisfaction of the Overall Adapted Quality (SQ) metric, we first evaluate the performance of the linearity of the UF model using the curve fitting. Then, we discuss the problem of overfitting and explain how we use the Cross-Validation (CV) methods to estimate more accurately the performance of the linearity of the UF model. Finally, we study the performance of 31 different polynomial models of degree up to 2, and discuss their results with respect to the linear model.

13.4.2.1 Evaluating the Performance of the Linearity of the Utility Function Model

As stated in Section 13.2.4, 5 metrics were rated by 20 users for each of the 18 adapted videos. In fact, these metrics were carefully chosen so that their values could be used as a subjective mapping of the UF and its parameters values (see Table 13.4). The affected area parameter is mapped to itself as it cannot be qualitatively measured.

TABLE 13.4 – Mapping between subjective and objective values.

Objective values of the parameters	**Subjective values of the metrics**
p_1 : affected area	p_1 : affected area
p_2 : affected priority area	Semantic Perceived Quality (SemPQ)
p_3 : spatial perceived quality	Spatial Perceived Quality (SPQ)
p_4 : temporal perceived quality	Temporal Perceived Quality (TPQ)
p_5 : Time efficiency	Time Efficiency (TE)

In particular, for each of the 18 adapted video pair $\langle V_i,\ M_j \rangle$, the observers rate the SQ of the adaptation plan generated by M_j over the video V_i. Thus, the performance of our utility function model is evaluated by depicting the relationship between the objective global utility values of the adaptation plans computed by the global utility function (refer to 10.1.3), and the actual MOS values of SQ. In order to calculate the objective global utility values of the adaptation plans, we need first to estimate the values of the weights w_i and the value of the decay constant b' for p_4.

	AffectedArea	SemPQ	TPQ	SPQ	TE	SQ
V1-M1	0.000	0.300	0.288	0.713	0.750	0.350
V1-M2	0.923	0.900	0.875	0.738	0.200	0.725
V1-M3	0.190	0.300	0.300	0.738	0.713	0.438
V1-M4	0.190	0.400	0.425	0.750	0.725	0.500
V1-M5	0.590	0.800	0.725	0.813	0.188	0.725
V1-M6	0.800	0.900	0.900	0.763	0.125	0.763
V2-M1	0.000	0.213	0.275	0.763	0.750	0.288
V2-M2	1.000	0.900	0.800	0.738	0.075	0.775
V2-M3	0.440	0.475	0.463	0.838	0.763	0.550
V2-M4	0.440	0.438	0.438	0.788	0.750	0.475
V2-M5	0.430	0.788	0.713	0.775	0.450	0.763
V2-M6	0.830	0.850	0.788	0.813	0.263	0.788
V3-M1	0.000	0.075	0.163	0.775	0.813	0.188
V3-M2	0.985	0.813	0.800	0.288	0.025	0.538
V3-M3	0.294	0.275	0.288	0.775	0.725	0.338
V3-M4	0.271	0.388	0.363	0.713	0.738	0.363
V3-M5	0.830	0.738	0.763	0.288	0.050	0.488
V3-M6	0.813	0.750	0.638	0.650	0.338	0.650

FIGURE 13.5 – MOS values of p_1, SemPQ, TPQ, SPQ, TE and SQ.

To begin with, for the 18 adapted videos pairs $\langle V_i,\ M_j \rangle$ and according to Equation 13.2, we calculate the MOS value of the re-scaled rating scores for the five metrics including the affected area parameter p_1 (see Figure 13.5).

Estimating the values of the weights w_i : in Section 9.2, we applied a linear model for the utility function, since it is usually assumed to be a good one. Based on this assumption, we estimate the values of the weight w_i by fitting the SQ values and the first five data columns (p_1 : affected area, SemPQ, TPQ, SPQ, TE) by the equation below :

$$SQ = w_1 \times p_1 + w_2 \times SemPQ + w_3 \times TPQ + w_4 \times SPQ + w_5 \times TE \qquad (13.5)$$

The curve fitting was done by the solver tool in Excel while minimizing the sum of squared differences between the actual SQ values and the values predicted by the right-hand side of Equation 13.5. It found the following values for the weights : $w_1 = 0.08$, $w_2 = 0.32$, $w_3 = 0.24$, $w_4 = 0.22$ and $w_5 = 0.14$. In order to evaluate the error during the fitting process, we calculate the RMSE. Its value is revealed to be very low $\simeq 0,0763$, thus indicating less prediction errors. In addition, we calculate the R-Squared R^2 to measure the precision of the curve fitting. The value of R^2 was found to be very high $\simeq 0,95$, thus indicating a good performance of the linear model.

Estimating the value of the decay constant b' for the spatial perceived quality parameter p_4 : as previously stated, for each adapted video pair $\langle V_i,\ M_j \rangle$, the observers rate the SQ of the adaptation plan generated by M_j over the video V_i. Having an estimation of the value of the decay constant b for p_3 and the values of the weights, the value of the decay constant b' can be found by fitting the SQ data with the global utility values of the adaptation plans. Indeed, for each pair $\langle V_i,\ M_j \rangle$, the global utility value of its resulting adaptation plan can be expressed in terms of the decay constant b'. Since the domain values of SQ is $[0, 1]$, we normalize the global utility values between $[0, 1]$. This is done by dividing the values computed by the global utility function, by the sum of the shots size that undergo the adaptation process. Therefore, by executing the curve fitting, the value of b' was estimated to 239.

Performance Evaluation of the Utility Function Model : The performance of our utility function model is evaluated by depicting the relationship between the values obtained from the global utility function and the actual MOS values of SQ. In total, we had 18 objective values to evaluate, such that $b = 2.8712$, $b' = 239$, $w_1 = 0.08$, $w_2 = 0.32$, $w_3 = 0.24$, $w_4 = 0.22$ and $w_5 = 0.14$. As stated in Section 13.4, we evaluate the ability of UF to estimate the MOS values with respect to the prediction accuracy, prediction monotonicity and errors evaluation (see Figure 13.6).

To begin with, we perform the t-test in Excel between the MOS of SQ and the global utility values. The size of the Pearson linear correlation coefficient (r) is found to be $r = 0.84$ with a $p - value$ equals to 0.012. These results imply that the correlation is statistically significant (as $p - value = 0.012 < 0.05$), and that the global utility function values highly correlates with the Mean Opinion Score (MOS) ratings of the human observers (as $r = 0.84$). Then, we calculate the Spearman rank-order correlation coefficient(r_s), which is generally responsible for providing an evaluation of the prediction monotonicity. The calculated value of $r_s = 0.79$ shows a high positive correlation. Finally, we calculate the RMSE to evaluate the errors of our UF model. The value of RMSE was found to be 0.13, which indicates a low prediction errors between the MOS of QS and the global utility function values. Therefore, it shows a good performance of UF model.

Though the result of the curve fitting shows a good performance of the UF model, training a model and evaluating its statistical performance on the same data may yield

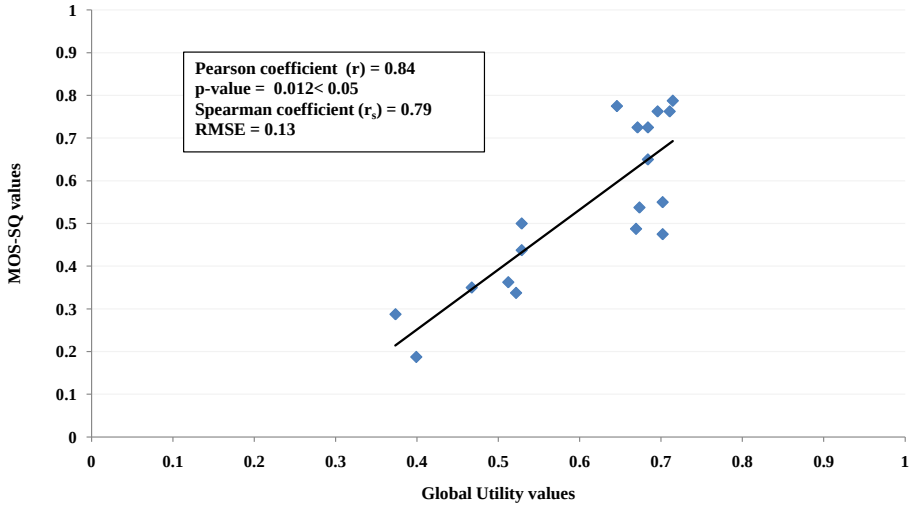


FIGURE 13.6 – Performance evaluation of the utility function.

an overoptimistic result [73]. For instance, given the UF model and the training data set to which the model can be fit, the curve fitting process find the optimal values for b, b' and w_i to make the model fit the training data set as well as possible. However, when presented with a new sample of independent data from the same population as the training data, this model would likely predict very poorly. This phenomenon is called overfitting, and is particularly likely to happen when the size of the training data set is small, or when the number of parameters in the model is large. To avoid this phenomenon, the idea is to split the available data set into a training set to which the model is fitted, and a test set that is used to evaluate its performance. One solution to overfitting involves Cross-Validation (CV) that estimates how well a model just learned from some training data is going to perform on future data. A general description of the CV strategy can be found in [41]. In the next section, we explain how we use the CV method to estimate more accurately the performance of the linearity of the UF model, and argue the selection of the linear model among 31 different ones.

13.4.2.2 Cross Validation Study to Select Utility Function Model

CV is a statistical method for validating a predictive model [10] [41]. As previously discussed, it was first introduced to fix the over-optimistic result derived from training a model and evaluating its statistical performance on the same dataset. Different types of CV methods can assist in the detection of overfitting (see Survey [10]).

Given the small size of our data set (18 values as depicted in Figure 13.5), we use the Leave-One-Out Cross-Validation (LOOCV) method that is known for making maximum use of the data set and for being feasible for small set. Indeed, Leave-One-Out (LOO) is an exhaustive CV method that learns and tests on all possible ways to split the original data set into a training and a test set. Given N data points in the data set, LOOCV "leaves out" one data point from the data set, trains on the remaining $N-1$ data points, and uses the "left out" data point to test the performance of the learned model on "new" data. This process is iterated on all the N data points of the data set and the average error is computed and used to evaluate the model. Besides the LOOCV method, we also developed three algorithms that use the Leave-One-label-Out Cross-Validation (LOLOCV) scheme, where each training set is constituted by all

the data points except the ones related to a specific label (e.g., specific user, specific method). The aim behind these algorithms is to estimate how accurately the UF model will perform in practice.

For the purpose of this thesis, we study the performance of different polynomial models of degree up to 2, and discuss their results with respect to the linear model one. Since we have at maximum 5 parameters (p_1, p_2, etc.) and their degree could be 1 or 2, thus we had a total of 32 different models, including the linear one, to be tested (see Figure 13.7).

	Polynomial degree of				
	p_1	p_2	p_3	p_4	p_5
model 1	1	1	1	1	1
model 2	1	1	1	1	2
model 3	1	1	1	2	1
model 4	1	1	1	2	2
model 5	1	1	2	1	1
model 6	1	1	2	1	2
model 7	1	1	2	2	1
model 8	1	1	2	2	2
model 9	1	2	1	1	1
model 10	1	2	1	1	2
...	...	...	...	...	...
model 30	2	2	2	1	2
model 31	2	2	2	2	1
model 32	2	2	2	2	2

FIGURE 13.7 – 32 polynomial models of degree up to 2.

Given our dataset, four algorithms were written in RStudio [14] in order to compare the performance of the 32 polynomial models. The polynomial function used in the fit is give as follows :

$$\begin{aligned} SQ \sim \; & poly(AffectedArea, degreeofp_1) + poly(SemPQ, degreeofp_2) + \\ & poly(TPQ, degreeofp_3) + poly(SPQ, degreeofp_4) + poly(TE, degreeofp_5) \end{aligned} \tag{13.6}$$

These algorithms are listed and explain in the following :

- **LOOCV-line :** given the 18 data points in the data set, the algorithm leaves out one data point from the data set, then finds the model that fits the remaining 17 data points by using Equation 13.6, and finally predicts the value of the left data point using the fitted model. This process is iterated on all the 18 data points.

- **Leave-One-Video-Out :** given the 18 data points in the data set, the algorithm first partitions the latter in 3 folds labelled by the 3 tested videos, whereas each fold will contain 6 data points since each video is adapted by 6 different adaptation methods. Then, the algorithm leaves out one fold from the data set (i.e., 6 data points in the test set related to a specific video), and trains the remaining

14. http://www.rstudio.com/

two folds (i.e., 12 values in the training set). Finally, the values of the left data points are predicted using the fitted model. This process is iterated on all the 3 folds.

- **Leave-One-Method-Out :** the algorithm uses the same process as the Leave-One-Video-Out, but instead it partitions the data set in 6 folds labelled by the 6 tested adaptation methods, whereas each fold will contain 3 data points as each adaptation method is applied on three videos.

- **Leave-One-User-Out :** given the 360 data points in the data set, that are the rating scores of the 20 users for 18 tested videos (see Table 13.3), the algorithm first partitions the data set in 20 folds labelled by the 20 tested users, whereas each fold contains 18 data points. Then, the algorithm leaves out one fold from the data set (i.e., 18 data points in the test set), and trains the remaining 19 folds (i.e., 342 values in the training set). Finally, the values of the left data points are predicted using the fitted model. This process is iterated on all the 20 folds.

For each of the 32 models and for each algorithm, we calculate the RMSE and the correlation of the model with the actual MOS values of SQ. We compare the performance of the 32 evaluated polynomial models based on these two criteria. The result of this evaluation is depicted in Table 13.5. For each criterion, we computed its variation range (i.e., Min. and Max.) as well as its value for the linear model.

TABLE 13.5 – Performance of the linear model for the UF among the 31 tested polynomial models

	RMSE			**Correlation**		
	Min.	*Max.*	*Linear model*	*Min.*	*Max.*	*Linear model*
LOOCV-line	0.039	0.062	0.039	0.942	0.977	0.977
Leave-One-Video-Out	0.052	0.698	0.052	0.177	0.962	0.962
Leave-One-Method-Out	0.041	0.091	0.041	0.880	0.976	0.975
Leave-One-User-Out	0.196	0.199	0.197	0.755	0.761	0.758

As can be observed from the table, the linear model is a good predictor of the QoE reported by the users. The predicted values show a strong linear correlation with the user ratings with a low RMSE. Though it is so simple, the linear model performs very well compared to the other 31 tested models, and sometimes it performs the best. Finally, by analysing the results of this evaluation, we noticed that the performance of the third and the fifth model (see Figure 13.7), is close to the one of the linear model. Due to the small scale of this experiment, it is not possible to decide whether one of them is clearly superior to the linear model.

13.4.3 One-Dimensional vs. Two-Dimensional Adaptation Methods

Figure 13.8 illustrates the obtained MOS value of the SQ metric with its associated CI, for each video V_i adapted by the method M_j.

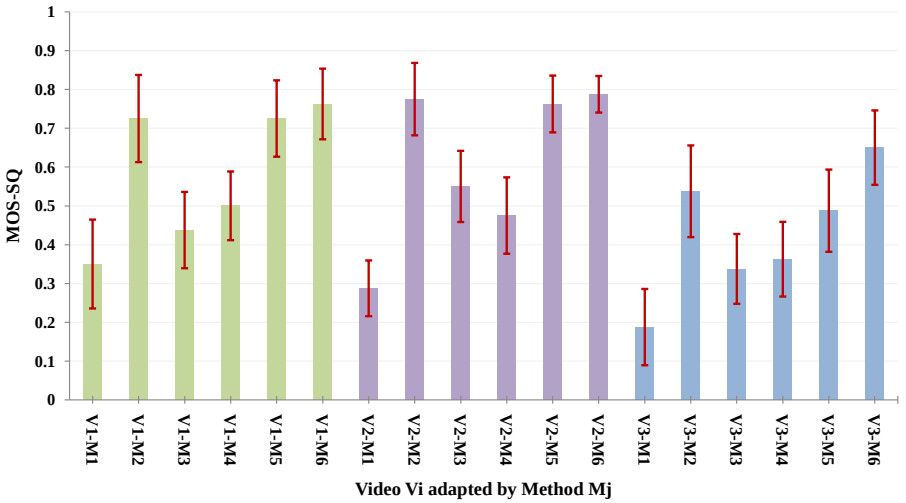


FIGURE 13.8 – The MOS values of the SQ for 18 adapted videos.

As can be observed from the figure, the overall satisfaction ratings of the videos adapted by a two-dimensional adaptation methods (i.e., V1-M5, V1-M6, V2-M5, V2-M6, V3-M5 and V3-M6) outperforms the ones adapted by most one-dimensional methods. However, if the size of the object to be adapted in the video is small, like in video V_1 and V_2, then the overall satisfaction for applying M_2 is close to M_5 and M_6. This is due to the definition of the UF model, which favors the $Drop - Object$ as an option, whenever the removal of an object does not result in high spatial artifacts, and its execution time is low. As stated in Section 9.2.4.2 and shown in F.3, this is the case of removing an object of small size.

13.5 Synthesis

In this chapter, we presented the results from the subjective video quality assessment tests, which we conducted to evaluate the performance of our Utility Function (UF) model and its parameters. The performance was determined from a comparison between viewer ratings obtained in controlled subjective tests and quality predictions from the UF model. For this purpose, we adopted the methodologies proposed by Recommendation ITU-RBT.500-13 [61] and ITU-T Recommendation P. 910, 2008 [60]. Furthermore, we evaluated the performance of the UF and its parameters by following the methodology suggested by the video quality experts group (VQEG) [44].

To begin with, we evaluated the performance of the temporal perceived quality parameter p_3 described in Section 9.2.4.1. The subjective assessments showed that the parameter p_3 is a good predictor of the user's QoE for a decay constant value b, equal to $2,871$. This value was found by fitting the exponential model of the parameter p_3 with the MOS values. The predicted ratings of p_3 has a correlation of 0.93 with the actual Mean Opinion Score (MOS) measured from subjective tests. The result of the t-test (i.e., $p - value = 0.002$) outlined the statistical significance of the correlation. Moreover, the value of the Root Mean Square Error (RMSE), that is 0.1608, showed that the p_3 model is a good predictor of the MOS for the Temporal Perceived Quality (TPQ) given by the users. These experimental results verified that the exponential decay model for the parameter p_3 is a good choice.

Then, we evaluate the performance of the linearity of the UF model and show that the linear model for the utility function is a good one among other models. As presented in Section 13.4.2.1, the subjective assessments demonstrate that the parameters of the UF model is a good predictor of the QoE reported by the users. After estimating the values of the weights and the decay constant b' of the parameter p_4, the predicted ratings by UF show a strong linear correlation of 0.84 with the actual Mean Opinion Score (MOS) measured from subjective tests. The result of the t-test (i.e., $p-value = 0.012$) outlined the statistical significance of the correlation. Moreover, the value of the Root Mean Square Error (RMSE), that is 0.13, showed that the UF model is a good predictor of the MOS for the overall satisfaction of the users with respect to the adaptation. Furthermore, in order to avoid to problem of overfitting, we used CV methods. The evaluation results showed that the values predicted by the linear model of the UF are highly correlated with the user ratings with a low RMSE. More important, these results also showed that the linear model is among the top three models when compared to 31 polynomial models.

Besides the validation of our proposed models and the evaluation of their performance, further conclusions were drawn from the subjective test results. We verified that the overall satisfaction ratings of the videos adapted by a two-dimensional adaptation methods performs the ones adapted by most one-dimensional methods. In the cases of removing an object of small size, we found that the overall satisfaction quality when only applying the $Drop - Object$ is close to the quality provided by two-dimensional adaptation methods. Moreover, based on the estimated values of the weights, we showed that each of the five parameters presented in the UF has an effect on the Quality of Experience (QoE). In particular, the affected priority area, spatial perceived quality and temporal perceived quality have a strong effect on the Quality of Experience (QoE), since their weighted values are high. Finally, since the weighted value of the processing cost is low, we concluded that most of the users prefer to trade off a higher waiting time for a better adaptation quality.

Despite the fact that the results are promising, we are aware that they are still preliminary and, more evaluations on a large scale for several users and videos are necessarily to validate the work in a conclusive way. Indeed, more evaluations are required to generate a more precise values of the weights, b and b'. Furthermore, intensive experimentations on a large scale will help us to prove if other models exist and they are competitive to the linear one.

Chapter 14

Conclusion and Future Work

In this chapter, we summarize the major contributions of this thesis, discusses some improvement of the PIAF framework, and highlight few possible directions for the future research.

14.1 Conclusion

In this thesis, we presented contributions related to Universal Multimedia Experience (UME), which is the notion that a user should receive informative adapted content anytime and anywhere. Personalization of videos, which adapts their content according to user preferences, is a vital aspect of achieving the UME vision. Realizing this personalization for UME is not an easy or trivial task. Indeed, it requires a fine-grained adaptation solution, which in turn requires joint consideration of several other issues : analysis of video content, understanding and modelling of user preferences, definition of quality models based on the concept of Quality of Experience (QoE), application of adaptation operations including the finest granularity (object level). In adaptation frameworks targeting UME, the most challenging task is that of the Adaptation Decision Taking Engine (ADTE). Indeed, for each entity undergoing the adaptation process, the ADTE must decide on the adequate adaptation operator that satisfies the user's preferences while maximizing his/her quality of experience. To achieve this, the ADTE must resolve three challenges. The first one is to objectively measure the quality of the adapted video, taking into consideration the multiple aspects of the QoE. The second challenge is to assess beforehand this quality in order to choose the most appropriate adaptation plan. The third challenge is to resolve conflicting or overlapping semantic constraints, in particular conflicts arising from constraints expressed by intellectual property rights owner. Although several adaptation frameworks exist in the literature, the semantic adaptation at the object level involving the removal adaptation operations type, is noticeably still an open issue (see Part I).

We tackled the aforementioned challenges by proposing a Utility Function (UF), which integrates semantic concerns with user's perceptual considerations. This function evaluates the utility of an adaptation according to five parameters : affected priority, affected priority area, spatial perceived quality, spatial perceived quality and processing cost. UF models the relationships among adaptation operations, user preferences, and the quality of the video content with a threefold purpose : 1) preserving the semantic integrity of the content by minimizing the overall impact of the adaptation especially on semantically critical parts of the content ; 2) maximizing the spatial and temporal

perceived quality of the adapted content ; and 3) minimizing the processing cost of the adaptation operation. The output of the UF is a utility matrix, which the ADTE uses to perform a multi-level piecewise reasoning to compute the adaptation plan that maximizes the user perceived quality. We also considered the effect of integrating intellectual property rights in the adaptation process. In some cases, the best adaptation plan satisfying the end-user constraints violates the owner's rights, thus resulting in conflicting constraints. This situation leads to an optimization problem, which we mapped to 0-1 Multiple-Choice Knapsack Problem (0-1 MCKP).

We also considered the problem of generating the context and content information required as input to our adaptation process. With respect to context information, we formalize the user constraints and proposed an extension to the MPEG-7 and MPEG-21 standards to include them under a new type of preferences called *UsageSemanticPreferences*. For content information, we designed a content model representing the spatial, temporal, and semantic information related to a video. The originality of this model lies in particular in its expressive representation of the spatial, temporal and semantic properties of video objects. Moreover, we developed a Semantic Video Content Annotation Tool (SVCAT) that assists annotators in generating video annotations according to the video model. The novelty of SVCAT lies in its automatic propagation of the object localization and description metadata realized by tracking their contour through the video, thus drastically alleviating the task of the annotators. Experimental evaluation shows that SVCAT is able to provide accurate metadata with nearly exact contours of multiple deformable objects.

All these contributions were carried out as part of a coherent and integrated framework called Personalized vIdeo Adaptation Framework (PIAF). PIAF is a complete modular MPEG standard compliant framework that covers the whole process of semantic video adaptation. Table 14.1- 14.2 provide a positioning of PIAF among the video adaptation frameworks KoMMa, CAIN, DCAF, DANAE, SAF and NinSuna presented in Chapter 4.

We validated the performance of PIAF with qualitative and quantitative evaluations, which assess the performance and the efficiency of the proposed adaptation decision-taking engine within the framework. The experimental results show that UF has a high correlation with subjective video quality evaluation. This was determined from a comparison between viewer ratings obtained in controlled subjective tests and quality predictions from the UF model. This analysis demonstrates that the UF model is a good predictor of the QoE reported by the users. The predicted ratings show a strong linear correlation of 0.84 with the Mean Opinion Score (MOS) ratings.

TABLE 14.1 – Positioning of PIAF among video adaptation frameworks.

	Granularity level		**Adaptation operation**		**Adaptation approach**
	considered levels	*object*	*considered operations*	*removal*	
koMMa [63]	video, frame	No	format transcoding, scaling	No	technical-driven
CAIN [82]	video, frame, object	Yes	format transcoding, scaling, substitution	No	technical-driven + semantic-driven
DCAF [123]	video	No	format transcoding, scaling	No	technical-driven
DANAE [106], SAF [149]	video, scene, shot, frame	No	format transcoding, scaling, substitution, selection	No	technical-driven + semantic-driven
NinSuna [133]	video, scene, shot	No	format transcoding, removal, selection	Yes	technical-driven + semantic-driven
PIAF [32]	**video, scene, shot, frame, object**	**Yes**	**removal**	**Yes**	**semantic-driven**

TABLE 14.2 – Positioning of PIAF among video adaptation frameworks (follow).

	ADT method	**QoE-awareness**		**Adaptation focus**	**Conflicting constraints**	**Owner rights**
		SemQ	*PQ*			
koMMa [63]	knowledge-based	No	No	terminal	No	No
CAIN-21 [82]	hybrid	No	Yes	terminal + user	No	No
DCAF [123]	quality-based	No	No	terminal	No	No
DANAE [106], SAF [149]	hybrid	No	Yes	terminal + user	No	No
NinSuna [133]	quality-based	No	No	terminal + user	No	No
PIAF [32]	**quality-based**	**Yes**	**Yes**	**user**	**Yes**	**Yes**

14.2 Future Work

This work could be developed in several directions, both in the areas of semantic content adaptation in PIAF and video annotation in SVCAT. In the following, we highlight some of the envisaged future work that can be considered as a continuation to this thesis.

14.2.1 Adaptation in PIAF

Below is a list of improvement to be concerned for the conception and evaluation of PIAF :

- **Using ontologies to enhance the interoperability in PIAF :** in this thesis, Classification Schemes (CS) are used to describe usage context (e.g., usage environments and users preferences) and video content. However, the interoperability of PIAF could be enhanced by building a unified and comprehensive ontology that can be used to describe semantic constraints related to specific domain.

- **Dealing with semantic constraints related to more than one object :** in the current implementation of PIAF, we only dealt with semantic constraints related to one single object within a shot. In practice, a constraint can involve more than one object. An interesting direction would be to extend the ADTE so that it considers several objects and their semantic relations when computing an adaptation plan, and treat cases where objects occlude or their associated SOFSs overlap. Furthermore, the current implementation of AEE does not include the development of all the adaptation tools. Only the tools related to the $Drop-Shot$ and $Drop-SOF$ were implemented. The inpainting tool for the $Drop-Object$ operation is still under progress. We only managed to make it work for specific shapes of objects. Though the implementation of the inpainting tool was out of the scope of this thesis, it is important to implement it, so that the adaptation during the subjective evaluations could be made online.

- **Conducting experimentation on a large scale for several users and videos :** though the results of the experimental evaluation conducted in this thesis are promising, doing intensive experimentation on a large scale for several users and videos is necessarily to validate the work in a conclusive way. Indeed, more evaluations are required to generate a more precise values of the weights and, of b and b' that are the constant decay of both temporal and perceived quality parameter. Moreover, doing intensive experimentation on a large scale enables us to find a range or class of the weights values instead of just fixing a values for each. For instance, if the user wants to make a trade off of lower quality for a lower waiting time, in this case the weight of the execution time parameter should be high and the one of the perception should be low. However, if the user accepts to make a trade off of higher quality for a higher waiting time, in this case the weight of the temporal and perceived quality parameter as well as the affected priority area should be higher than the other parameters.

Furthermore, important future developments of this work include but not limited to :

- **Optimizing the video adaptation process :** a possibility would be to make use of adapted segments. As the object removal operation is computationally intensive, it makes sense to store the adapted segments and reuse it whenever it is requested again. This not only minimizes the overall computational cost but also reduces the time users need to wait to receive the adapted video. In distributed video delivery systems, this approach can be exploited even more by storing adapted segments on proxy servers- computers that are kept closer to the user and assumed to have a high transmission rate and low latency connection to the users. Basically, each proxy can be made to store object level adapted segments of popular videos requested frequently by its clients. The proxies can also exchange these adapted segments, via some sort of cooperation scheme. This reduces the load on the server and backbone network while increasing availability of adapted segments at lower storage cost. Such a strategy can be implemented without requiring much change to the infrastructure or algorithms that are used in distributed video delivery. We just need to have a mechanism that gives more importance to object level adapted segments of popular videos during purging proxy caches. This requires a slight modification to the cache replacement algorithms used in the proxies. We argue it is worth doing this given the fact that such an approach facilitates the video delivery by minimizing the extra video startup delay that would have occurred due to the time required for adaptation.

  In another realm, compressed domain video adaptation can be considered as a means to minimize computational requirement of the object removal operation. In compressed domain video adaptation, the adaptation is done based on high-level structural description of the video expressed in XML. This technique has been recently applied for spatial and temporal video adaptation [149], as it is computationally less expensive due to the reality that the cascaded decode-process-encode operations are avoided. Generic Bitstream Syntax Schema (gBS Schema) offered by MPEG-21 DIA enables to describe the high-level structure of the bitstream using XML, and to create a generic Bitstream Syntax Description (gBSD) document. The gBSD makes it possible to describe the bitstream in a coding format-agnostic manner and facilitates the adaptation to be done in the XML domain. Object-based adaptation operations can be facilitated by creating a mapping between video slices and objects in the video, and inserting this information in the gBSD. For instance, when re-encoding a video entering our system, the area covered by the object (based on the size and location information obtained from the MPEG-7 descriptions) is approximated to a rectangular shape and all slices in the area are indicated to represent that object. This can be achieved in advanced video standards such as H.264 by modifying the basic encoders used in the video encoding process.

- **Deployement of PIAF on the Web :** another important future development of this work is to deploy it on the Web. To this end, we envision an architectural approach similar to the one defined by Prangl et al. [103], in which the ADTE is implemented as a Web service, which can be easily combined with a Web server responsible for managing user inputs (queries, profile information specification). This approach will enable a smoother integration into existing Web frameworks. In the longer range, we can strive to extend our approach of basing our work on technology by using W3C Web standards in addition to MPEG standards.

This is important in particular due to the current works towards a media aware semantic Web resolving the discrepancy between multimedia metadata formats and the semantic Web [135], which materializes at W3C by the Video in the Web activity. This activity is interesting as it brings together many experts of the research community in this domain, and is thus a good basis to anticipate future developments. This is particularly the case for the issue of providing multimedia metadata on the Web in a structured way, which is targeted by the Ontology for Media Annotations [15]. A stated goal of this activity is to define a core vocabulary on which the descriptions generated according to a large number of multimedia metadata formats may be mapped. Moreover, the current version of the standard provides an implementation of the ontology that is compatible with the semantic Web, using RDF/OWL. The ontology itself is highly compatible with the approach presented in this thesis. Indeed, the MPEG-7 mapping covers most tags required by our annotation model such as the "decomposition", temporal and spatial tags required for the definition of the scenes, shots and objects, and the tags for textual descriptions that enable us to describe objects using keywords. More precisely, when it comes to the "decomposition" tags, they are mapped to Uniform Resource Identifiers (URIs) defined in the "Media Fragments URI" specification [16], thereby enabling us to benefit from the results of another important ongoing activity of the W3C Video on the Web working group. To summarize, our approach is fundamentally compatible with current Web standards, thus allowing for the deployment of PIAF on today's Web. Moreover, the current trend towards a tighter integration between multimedia metadata standards and the semantic Web clearly indicates that future developments of the standards will facilitate and increase the integration level of PIAF with other services that are based on Web standards.

– **Investigating the usefulness of saliency maps and motion maps in predicting the spatial perceived quality :** in this thesis, we proposed a metric (denoted p_4) to predict the spatial perceived quality by measuring the size of the spatial artifacts left by imperfect inpainting algorithms. We considered that the more spatial artifacts are resulting from imperfect inpainting, the more attention is drawn to these artifacts and the more the quality of perception is affected negatively. Though the experimentations prove the usefulness of this simple model, it probably leaves room for improvement.
Indeed, a number of studies have shown that under certain circumstances, very large changes can be made in a picture without being noticed by the observers [109] [117]. For instance, the artifacts of an inpainted object may go unnoticed. Indeed, the amount of information coming down the optic nerve - estimated to be in the range of 10^8 - 10^9 bits per second - far exceeds what the brain is capable of fully processing and assimilating into conscious experience [59]. A natural strategy called 'selective visual attention' was designed to deal with this biological bottleneck. This strategy consists in two steps : 1) selecting the portions of the input to be processed preferentially, that is selecting relevant or salient objects ; and 2) shifting the processing focus from one location to another in a serial fashion, that is shifting from the most relevant object to the less relevant one.
Several attention visual models, often referred to as saliency models, were developed in the literature [58] [75] [88]. These models take a scene as an input, and

15. http://www.w3.org/TR/mediaont-10/
16. http://www.w3.org/2008/WebVideo/Fragments/WD-media-fragments-spec/

compute the saliency of contained areas or objects based on their visual features such as color, edge orientation, luminance, etc. They provide a mean to objectively attribute values to such areas in a way that applications can predict human fixations and focus changes. Based on the saliency values given by these models, we think it is worthy to investigate to which extent the saliency of a surrounding or neighboring region of a removed object have an impact on the perceptual quality. For instance, we are wondering if the artifact left by removing an object (e.g., apple logo) within an image whereas the object is related to a remaining salient one (e.g., back of a computer), would catch more the attention of the user than if it was related to a none salient one. Furthermore, as the motion is known to be one of the most important visual attractors [75] [57], we are wondering if the artifact left by removing a static object would less attract the user attention than removing a moving object. Similar question could be investigated with respect to the motion of the objects, which are related or neighbours of the removed object.

14.2.2 Annotation in SVCAT

SVCAT can be improved to the point of being a complete interoperable annotation tool by supporting ontologies for semantic metadata in addition to Classification Schemes. Furthermore, the process of the object detection in SVCAT can be fully automated for domain-specific applications [33]. This requires having a training set that covers many variations of the object appearance. This training set could be created by extending the functionality of SVCAT so that it can learn from existing annotations. This would enable the automatic detection of the spatio-temporal location of objects in new videos using object recognition techniques. Clearly, these improvements of SVCAT can drastically reduce the effort required from the annotator.

Bibliography

[1] Multimedia Group . *VIA : Video Image Annotation Tool* **[online]**. Available : <mklab.iti.gr/project/via> (accessed 10.01.2013).

[2] Abebe Getnet, Coquil David, Kosch Harald. *Enhancing Semantic Video Adaptation Speed through Compressed Domain Adaptation.* **In :** Proc. of the 2011 Workshop on Multimedia on the Web (MMWEB '11), sept. 2011, Graz, Austria. IEEE Computer Society, 2011, pp 43–44.

[3] Adzic Velibor, Kalva Hari, Furht Borko. *A Survey of Multimedia Content Adaptation for Mobile Devices.* Journal of Multimedia Tools and Applications, 2011, vol. 51, n°1, pp. 379-396.

[4] Ahmed Rakib, Karmakar Gour C., Dooley Laurence S. *Region-Based Shape Incorporation for Probabilistic Spatio-Temporal Video Object Segmentation.* **In :** proc. of the IEEE International Conference on Image Processing (ICIP '06), oct. 2006, Atlanta, GA, USA. IEEE Computer Society, 2006, pp 2445–2448.

[5] Albanese Massimiliano, Chianese Angelo, Moscato Vincenzo, Sansone Lucio. *A Formal Model for Video Shot Segmentation and its Application via Animate Vision.* Journal of Multimedia Tools and Applications, 2004, vol. 24, n°3, pp. 253-272.

[6] Allen James F. *Maintaining Knowledge about Temporal Intervals.* Journal of Communications of the ACM, 1983, vol. 26, n°11, pp. 832-843.

[7] Anderson Greg, Ferro David, Hilton Robert. *Numbering Systems and Data Representations.* **In :** Connecting with Computer Science. 2^{nd} edition, Canada, Cengage Learning, jan. 2010, pp 248–275.

[8] Anscombe Francis J. *Graphs in Statistical Analysis.* Journal of American Statistician, 1973, vol. 27, n°1, pp. 17-21.

[9] ANSI T1.TR.74-2001 . *Objective Video Quality Measurement Using a Peak-Signal-to-Noise-Ratio (PSNR) Full Reference Technique.* Draft Technical Report T1.TR.74-2001 **[online]**. American National Standards Institute, Ad Hoc Group on Video Quality Metrics, 2001.).

[10] Arlot Sylvain, Celisse Alain. *A Survey of Cross-validation Procedures for Model Selection.* Journal of Statistics Surveys, 2010, vol. 4, pp. 40-79.

[11] Beauregard Russell, Corriveau Philip. *User Experience Quality : A Conceptual Framework for Goal Setting and Measurement.* **In :** Digital Human Modeling. Springer Berlin Heidelberg, jul. 2007, volume 4561 of *Lecture Notes in Computer Science*, pp 325–332.

[12] Berhe Hagos Girma. *Accès et Adaptation de Contenus Multimédia pour les Systèmes Pervasifs.* Thèse informatique. Lyon : INSA de Lyon, 2006, 305 p.

[13] Berhe Hagos Girma, Brunie Lionel, Pierson J-M. *Planning-based Multimedia Adaptation Services Composition for Pervasive Computing.* **In :** Proc. of the 5^{th} International Conference on Signal-Image Technology & Internet-Based Systems (SITIS '2009), dec. 2009, Marrakesh, Morocco. IEEE Computer Society, 2009, pp 326–331.

[14] Bordwell David, Thompson Kristin. *Film Art : An Introduction.* McGraw-Hill Humanities/Social Sciences/Languages, 10^{th} edition, jul. 2004, 544 p.

[15] Bruyne Sarah, Hosten Peter, Concolato Cyril, Asbach Mark, Cock Jan, Unger Michael, Feuvre Jean, Walle Rik. *Annotation Based Personalized Adaptation and Presentation of Videos for Mobile Applications.* Journal of Multimedia Tools and Applications, 2011, vol. 55, n°2, pp. 307-331.

[16] Burnett Ian S., Pereira Fernando, Van de Walle Rik, Koenen Rob. *The MPEG-21 Book.* Wiley, 1^{st} edition, apr. 2006, 462 p.

[17] Chan T. F., Vese L. A. *An Active Contour Model without Edges.* IEEE Transactions on Image Processing, 2001, vol. 10, n°2, pp. 266–277.

[18] Chan Tony F., Kang Sung Ha. *Error Analysis for Image Inpainting.* Journal of Mathematical Imaging and Vision, 2006, vol. 26, n°1-2, pp. 85-103.

[19] Chandrashekar Abhijith, Chandrashekar Priyanka, Mishra Soumya, Dambal Praveen. *User Perception of Quality of Distributed Multimedia.* Term Paper CIS6930 **[online]**. University of Florida (UFL), Departement of Computer & Information Science & Engineering (CISE), 2010. Available : <http://www.cise.ufl.edu/~phdambal/DMS-TermPaper.pdf> (accessed 04.05.2013).

[20] Chang Shih-Fu, Vetro Anthony. *Video Adaptation : Concepts, Technologies, and Open Issues.* Proceedings of the IEEE, 2005, vol. 93, n°1, pp. 148-158.

[21] Chen Chien-nan, Chu Cing-yu, Yeh Su-ling, Chu Hao-hua, Huang Polly. *Measuring the Perceptual Quality of Skype Sources.* ACM SIGCOMM Computer Communication Review, 2012, vol. 42, n°4, pp. 521-526.

[22] Cheng Wen-Huang. *A Semantic Framework for Object-Based and Event-Based Video Content Adaptation.* Phd thesis. Taiwan : Institute of Networking and Multimedia, National Taiwan University, 2008, 172 p.

[23] Comaniciu Dorin, Meer Peter, Member Senior. *Mean Shift : A Robust Approach toward Feature Space Analysis.* Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, vol. 24, n°5, pp. 603-619.

[24] DANAE Project . *DANAE : Dynamic and distributed Adaptation of scalable multimedia coNtent in a context-Aware Environment* **[online]**. Available : <http://danae.rd.francetelecom.com/> (accessed 03.05.2013).

[25] Dasiopoulou Stamatia, Giannakidou Eirini, Litos Georgios, Malasioti Polyxeni, Kompatsiaris Yiannis. *A Survey of Semantic Image and Video Annotation Tools.* **In :** Knowledge-driven multimedia information extraction and ontology evolution. Springer Berlin Heidelberg, 2011, number 6050, pp 196–239.

[26] Davis Fred D., Bagozzi Richard P., Warshaw Paul R. *User Acceptance of Computer Technology : a Comparison of Two Theoretical Models.* Journal of Management Science, aug. 1989, vol. 35, n°8, pp. 982-1003.

[27] De Cock Jan, Notebaert Stijn, Vermeirsch Kenneth, Lambert Peter, Van de Walle Rik. *Dyadic Spatial Resolution Reduction Transcoding for H.264/AVC.* Journal of Multimedia Systems, 2010, vol. 16, n°2, pp. 139-149.

[28] Dijkstra Edsger W. *A Note on Two Problems in Connexion with Graphs.* Journal of Numerische Mathematik, 1959, vol. 1, n°1, pp. 269-271.

[29] Dudziński Krzysztof, Walukiewicz Stanislaw. *Exact methods for the knapsack problem and its generalizations.* European Journal of Operational Research, 1987, vol. 28, n°1, pp. 3-21.

[30] Dyer M.E., Kayal N., Walker J. *A Branch and Bound Algorithm for Solving the Multiple-choice Knapsack Problem.* Journal of Computational and Applied Mathematics, 1984, vol. 11, n°2, pp. 231-249.

[31] El-Khoury Vanessa, Bennani Nadia, Coquil David. *Utility Function for Semantic Video Content Adaptation.* **In :** Masters and Doctoral Colloquium (MDC) in conjunction with IIWAS '10, dec. 2010, Paris, France. ACM, 2009, pp 921–924.

[32] El-Khoury Vanessa, Coquil David, Bennani Nadia, Brunie Lionel. *Personalized vIdeo Adaptation Framework (PIAF) : High-Level Semantic Adaptation.* Journal of Multimedia Tools and Applications, 2012.

[33] El-Khoury Vanessa, Jergler Martin, Abebe Bayou Getnet, Coquil David, Kosch Harald. *Fine-granularity Semantic Video Annotation : An Approach based on Automatic Shot Level Concept Detection and Object Recognition.* International Journal of Pervasive Computing and Communications, 2013, vol. 9, n°3, pp. 243-269.

[34] El-Khoury Vanessa, Jergler Martin, Coquil David, Kosch Harald. *Semantic Video Content Annotation at the Object Level.* **In :** Proc. of the 10^{th} International Conference on Advances in Mobile Computing & Multimedia (MoMM '12), dec. 2012, Bali, Indonesia. ACM, 2012, pp 179–188.

[35] Feiten Bernhard, Wolf Ingo, Oh Eunmi, Seo Jeongil, Kim Hae-Kwang. *Audio Adaptation According to Usage Environment and Perceptual Quality Metrics.* IEEE Transactions on Multimedia, 2005, vol. 7, n°3, pp. 446-453.

[36] Fisher Robert, Perkins Simon, Walker Ashley, Wolfart Erik. *HIPR : Hypermedia Image Processing Reference.* John Wiley and Sons, 1996, 318 p.

[37] Francisco-Revilla Luis. *A Picture of Hypervideo Today* **[online]**. Available : <http://www.csdl.tamu.edu/~l0f0954/academic/cpsc610/p-3.htm> (accessed 19.01.2013).

[38] Fu King-Sun, Mui J.K. *A survey on image segmentation.* Journal of Pattern Recognition, 1981, vol. 13, n°1, pp. 3-16.

[39] Furht Borko. *Encyclopedia of Multimedia.* Springer, USA, 2008, 1001 p.

[40] Gabriel Panis, Hutter Andreas, Heuer Jörg, Hellwagner Hermann, Kosch Harald, Timmerer Christian, Devillers Sylvain, Amielh Myriam. *Bitstream syntax description : a tool for multimedia resource adaptation within MPEG-21.* Journal of Signal Processing : Image Communication, 2003, vol. 18, n°8, pp. 721-747.

[41] Geisser Seymour. *The Predictive Sample Reuse Method with Applications.* Journal of the American Statistical Association, 1975, vol. 70, n°350, pp. 320-328.

[42] Ghinea Gheorghita, Thomas Johnson P. *Quality of Perception : User Quality of Service in Multimedia Presentations.* IEEE Transactions on Multimedia, 2005.

[43] Grill-Spector Kalanit, Rafael Malach. *The Human Visual Cortex.* Annual Review of Neuroscience, 2004, vol. 27, n°1, pp. 649-677.

[44] Group VQEG : Video Quality Experts. *Final Report From the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment.* Technical report **[online]**. 2000. Available : <http://csce.uark.edu/~jgauch/library/Video-Quality/index.html> (accessed 28.03.2013).

[45] Hanjalic Alan, Xu Li Qun. *Affective Video Content Representation and Modeling.* IEEE Transactions on Multimedia, 2005, vol. 7, n°1, pp. 143-154.

[46] Haralick Robert M., Shapiro Linda G. *Image Segmentation Techniques.* Journal of Computer Vision, Graphics, and Image Processing, jun. 1985, vol. 29, n°1, pp. 100-132.

[47] Heggland Jon. *OntoLog : Flexible Management of Semantic Video Content Annotations.* Phd thesis. Trondheim : Norwegian University of Science and Technology, 2005, 254 p.

[48] Heidelberger Michael, Klohr Cynthia. *Nature From Within : Gustav Theodor Fechner and His Psychophysical Worldview.* University of Pittsburgh Press, 1^{st} edition, feb. 2004, 456 p.

[49] Hinkle Dennis.E., Wiersma William, Jurs Stephen G. *Applied Statistics for the Behavioral Sciences*, volume 663. Houghton Mifflin, 2003.

[50] Hong-ying ZHANG, Qi-Cong . *A Survey on Digital Image Inpainting.* Journal of Image and Graphics, 2007, vol. 12, n°1, pp. 1-10.

[51] Hoßfeld Tobias, Hock David, Tran-Gia Phuoc, Tutschku Kurt, Fielder Markus. *Testing the IQX Hypothesis for Exponential Interdependency between QoS and QoE of Voice Codecs iLBC and G.711.* Technical report **[online]**. Würzburg University, Department of Distributed Systems, Währinger Straße 29, 1090 Wien, 2008. Available : <http://eprints.cs.univie.ac.at/375/> (accessed 24.03.2013).

[52] ISO/IEC . *Information Technology – Multimedia Content Description Interface – Part 5 : Multimedia Description Schemes.* ISO/IEC 15938-5. 2003, 730 p.

[53] ISO/IEC . *Information Technology – Multimedia Framework (MPEG-21) – Part 2 : Digital Item Declaration.* ISO/IEC 21000-2. 2005, 88 p.

[54] ISO/IEC . *Information Technology – Multimedia framework (MPEG-21) – Part 10 : Digital Item Processing.* ISO/IEC 21000-10. 2006, 121 p.

[55] ISO/IEC . *Information Technology – Multimedia framework (MPEG-21) – Part 7 : Digital Item Adaptation.* ISO/IEC 21000-7. 2007, 420 p.

[56] ISO/IEC . *Information Technology – Coding of Audio-visual Objects – Part 10 : Advanced Video Coding.* ISO/IEC 14496-10. 2010.

[57] Itti Laurent. *Quantifying the Contribution of Low-level Saliency to Human Eye Movements in Dynamic Scenes.* Visual Cognition, 2005, vol. 12, n°6, pp. 1093-1123.

[58] Itti Laurent, Koch Christof. *Comparison of Feature Combination Strategies for Saliency-based Visual Attention Systems.* **In :** Electronic Imaging'99. International Society for Optics and Photonics, 1999, pp 473–482.

[59] Itti Laurent, Koch Christof. *Computational Modelling of Visual Attention.* Journal of Nature Reviews Neuroscience, 2001, vol. 2, n°3, pp. 194-203.

[60] (ITU) International Communication Union. *Subjective Video Quality Assessment Methods for Multimedia Applications.* ITU-T Recommendation P.910. 2008, 42 p.

[61] (ITU) International Communication Union. *Methodology for the Subjective Assessment of the Quality of Television Pictures.* Broadcasting service (television), itu-r bt.500-13. 2012, 46 p.

[62] Jaimes Alejandro, Chang Shih-fu. *A Conceptual Framework for Indexing Visual Information at Multiple Levels.* **In :** Proc. of the IS&T/SPIE Internet Imaging, volume 3964, jan. 2000, San Jose, CA, USA. SPIE, 2000, pp 2–15.

[63] Jannach Dietmar, Leopold Klaus. *Knowledge-based Multimedia Adaptation for Ubiquitous Multimedia Consumption*. Journal of Network and Computer Applications, 2007, vol. 30, n°3, pp. 958-982.

[64] Jannach Dietmar, Leopold Klaus, Timmerer Christian, Hellwagner Hermann. *A knowledge-based Framework for Multimedia Adaptation*. Journal of Applied Intelligence, 2006, vol. 24, n°2, pp. 109-125.

[65] Johanneum Research . *SVAS : Semantic Video Annotation Suite* **[online]**. Available : <http://www.joanneum.at/digital/produkte-loesungen/semantic-video-annotation.html> (accessed 08.01.2013).

[66] Kass Michael, Witkin Andrew, Terzopoulos Demetri. *Snakes : Active contour models*. Int. Journal of Computer Vision, 1988, vol. 1, n°4, pp. 321-331.

[67] Kasutani Eiji. *New Frontiers in Universal Multimedia Access*. ITS Report LTS-REPORT-2004-019 **[online]**. Ecole Polytechnique Fédéral de Lausanne, 2004. Available : <http://infoscience.epfl.ch/record/87058> (accessed 03.02.2013).

[68] Kellerer Hans, Pferschy Ulrich, Pisinger David. *Knapsack Problems*. Springer Verlag GMBH, dec. 2003, 568 p.

[69] Kimiaei Asadi Mariam. *Adaptation de Contenu Multimédia avec MPEG-21 : Conversion de Ressources et Adaptation Sémantique de Scènes*. Thèse informatique et réseaux. Paris : Ecole Nationale Supérieure des Télécommunications, 2005, 222 p.

[70] Kipp M. *Anvil - A Generic Annotation Tool for Multimodal Dialogue*. **In :** Pro. of the 7^{th} Europ. Conf. on Speech Communication and Technology (Eurospeech 2001 Scandinavia), sep. 2001, Aalborg, Denmark.

[71] Lachner J., Lorenz A., Reiterer B., Zimmermann A., Hellwagner H. *Challenges Toward User-Centric Multimedia*. **In :** Proc. of the 2^{nd} International Workshop on Semantic Media Adaptation and Personalization (SMAP' 2007), dec. 2007, London, UK. IEEE Computer Society, 2007, pp 159–164.

[72] Lankton Shawn. *Sparse Field Methods*. Technical report **[online]**. Georgia Institute of Technology, 2009. Available : <http://www.shawnlankton.com/wp-content/uploads/articles/lankton-sfm-TR-2009.pdf> (accessed 22.06.2012).

[73] Larson Selmer C. *The Shrinkage of the Coefficient of Multiple Correlation*. Journal of Educational Psychology, 1931, vol. 22, n°1, pp. 45.

[74] Lausberg H., Sloetjes H. *Coding Gestural Behavior with the NEUROGES-ELAN System*. Journal of Behavior Research Methods, Instruments, & Computers, 2009, vol. 41, n°3, pp. 841-849.

[75] Le Meur Olivier, Le Callet Patrick, Barba Dominique. *Predicting visual fixations on video based on low-level visual features*. Vision Research, 2007, vol. 47, n°19, pp. 2483-2498.

[76] Leopold Klaus, Jannach Dietmar, Hellwagner Hermann. *A knowledge and Component based Multimedia Adaptation Framework*. **In :** Proc. of the IEEE 6^{th} International Symposium on Multimedia Software Engineering, dec. 2004, Miami, USA. IEEE Computer Society, 2004, pp 10–17.

[77] Levin A., Viola P., Freund Y. *Unsupervised Improvement of Visual Detectors using co-Training*. **In :** Proc. of the IEEE 9^{th} International Conference on Computer Vision, volume 1, oct. 2003, Nice, France. IEEE Computer Society, 2003, pp 626–633.

[78] Li Chung-Sheng, Mohan R., Smith J. R. *Multimedia Content Description in the Infopyramid*. **In :** Proc. of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, volume 6, may 1998, Washington, USA.

[79] Lin Weisi, Jay Kuo C-C. *Perceptual Visual Quality Metrics : A Survey*. Journal of Visual Communication and Image Representation, 2011, vol. 22, n°4, pp. 297-312.

[80] LIRIS laboratory (UMR 5205 CNRS) . *ADVENE : Annotate Digital Video, Exchange on the NEt (ADVENE Project)* **[online]**. Available : <http://liris.cnrs.fr/advene/> (accessed 03.01.2013).

[81] López Fernando, Jannach Dietmar, Martínez José M, Timmerer Christian, Hellwagner Hermann, García Narciso. *Multimedia Adaptation Decisions Modelled as non-Deterministic Operations*. **In :** Proc. of the 9^{th} International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '08), may 200, Klagenfurt, Austria. IEEE Computer Society, 2008, pp 46–49.

[82] López Fernando, Martínez José María, García Narciso. *CAIN-21 : An Extensible and Metadata-Driven Multimedia Adaptation Engine in the MPEG-21 Framework*. **In :** Semantic Multimedia. Graz, Austria, Springer Berlin Heidelberg.

[83] López Fernando, Martínez José María, García Narciso. *Automatic Adaptation Decision Making using an Image to Video Adaptation Tool in the MPEG-21 Framework*. **In :** Proc. of the 10^{th} Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '2009), may 2009, London, United Kingdom. IEEE Computer Society, 2009, pp 222–225.

[84] López Fernando, Nur G., Dogan S., Arachchi H.K., Mrak M., Martínez José María, García Narciso, Kondoz A. *Improving Scalable Video Adaptation in a Knowledge-based Framework*. **In :** Proc. of the 11^{th} International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '2010), apr. 2010, Lombardy, Italy. IEEE Computer Society, 2010, pp 1–4.

[85] Magalhães João, Pereira Fernando. *Using MPEG Standards for Multimedia Customization*. Signal Process : Image Communication, feb. 2004, vol. 19, n°5, pp. 437-456.

[86] Mahalingam Vijay Venkatesh. *Digital Inpainting Algorithms and Evaluation*. Phd thesis. Kentucky : University of Kentucky, 2010, 128 p.

[87] Manjunath B. S., Salembier Philippe, Sikora Thomas. *Introduction to MPEG-7 : Multimedia Content Description Interface*. John Wiley & Son, 1 edition, apr. 2002, 396 p.

[88] Marat Sophie, Phuoc Tien Ho, Granjon Lionel, Guyader Nathalie, Pellerin Denis, Guérin-Dugué Anne. *Modelling Spatio-temporal Saliency to Predict Gaze Direction for Short Videos*. International Journal of Computer Vision, 2009, vol. 82, n°3, pp. 231-243.

[89] Martínez José M. *MPEG-7 Overview (version 10)*. Report ISO/IEC JTC1/SC29/WG11N6828 **[online]**. International Organisation for Standardisation, oct. 2004. Available : <http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm#E11E10> (accessed 20.10.2012).

[90] Megino F.B., Sanchez José María Martínez, López V.V. *Virtual Camera Tools for an Image2Video Application*. **In :** Proc. of the 9^{th} International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '08), may 2008, Klagenfurt, Austria. IEEE Computer Society, 2008, pp 223–226.

[91] Money Arthur G, Agius Harry. *Video Summarisation : A Conceptual Framework and Survey of the State of the Art.* Journal of Visual Communication and Image Representation, 2008, vol. 19, n°2, pp. 121-143.

[92] Mu Mu, Romaniak Piotr, Mauthe Andreas, Leszczuk Mikolaj, Janowski Lucjan, Cerqueira Eduardo. *Framework for the Integrated Video Quality Assessment.* Journal of Multimedia Tools and Applications, 2012, vol. 61, n°3, pp. 787-817.

[93] Mukherjee Debargha, Delfosse Eric, Kim Jae-Gon, Wang Yong. *Optimal Adaptation Decision-taking for Terminal and Network Quality-of-Service.* IEEE Transactions on Multimedia, 2005, vol. 7, n°3, pp. 454-462.

[94] Nam Jeho, Ro Yong Man, Huh Youngsik, Kim Munchurl. *Visual Content Adaptation According to User Perception Characteristics.* IEEE Transactions on Multimedia, 2005, vol. 7, n°3, pp. 435-445.

[95] Nielsen Frank, Nock Richard. *Interactive Point-and-Click Segmentation for Object Removal in Digital Images.* **In :** Computer Vision in Human-Computer Interaction. Beijing, China, Springer Berlin Heidelberg, oct. 2005, volume 3766 of *Lecture Notes in Computer Science*, pp 131–140.

[96] Ohto Hidetaka, Hjelm Johan. *CC/PP Exchange Protocol based on HTTP Extension Framework* **[online]**. Available : <http://www.w3.org/TR/NOTE-CCPPexchange> (accessed 21.01.2013).

[97] Osher Stanley, Sethian James A. *Fronts Propagating with Curvature-dependent Speed : Algorithms based on Hamilton-Jacobi Formulations.* Journal OF Computational Physics, nov. 1988, vol. 79, n°1, pp. 12-49.

[98] Pastrana-Vidal Ricardo R., Gicquel Jean Charles, Colomes Catherine, Cherifi Hocine. *Sporadic frame dropping impact on quality perception.* **In :** Proc. of the SPIE 5292, Human Vision and Electronic Imaging IX, volume 5292, jun. 2004, San Jose, CA, USA. SPIE, 2004, pp 182–193.

[99] Pereira Fernando, Burnett Ian S. *Universal Multimedia Experiences for Tomorrow.* IEEE Signal Processing Magazine, 2003, vol. 20, n°2, pp. 63-73.

[100] Pinson Margaret H., Wolf Stephen. *A New Standardized Method for Objectively Measuring Video Quality.* IEEE Transactions on Broadcasting, sep. 2004, vol. 50, n°3, pp. 312-322.

[101] Pisinger David. *A minimal algorithm for the multiple-choice knapsack problem.* European Journal of Operational Research, 1995, vol. 83, n°2, pp. 394-410.

[102] Pradhan Sujeet, Tajima Keishi, Tanaka Katsumi. *A Query Model to Synthesize Answer Intervals from Indexed Video Units.* IEEE Transaction on Knowledge and Data Engeneering, sep 2001, vol. 13, n°5, pp. 824-838.

[103] Prangl Martin, Kofler Ingo, Hellwagner Hermann. *An MPEG-21-driven Utility-based Multimedia Adaptation Decision Taking Web Service.* **In :** Proc. of the 1^{st} International Conference on Ambient Media and Systems, feb. 2008, Quebec, Canada. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp 1–8.

[104] Prangl Martin, Szkaliczki Tibor, Hellwagner Hermann. *A Framework for Utility-based Multimedia Adaptation.* IEEE Transactions on Circuits and Systems for Video Technology, jun. 2007, vol. 17, n°6, pp. 719-728.

[105] Press Niso. *Understanding Metadata.* Report **[online]**. American National Standards Institute, 2004. Available : <http://www.niso.org/publications/press/UnderstandingMetadata.pdf> (accessed 10.10.2012).

[106] Ransburg Michael, Hellwagner Hermann, Cazoulat Renaud, Pellan Benoit, Concolato Cyril, De Zutter Saar, Poppe Chris, Van de Walle Rik, Hutter Andreas. *Dynamic and Distributed Adaptation of Scalable Multimedia Content in a Context-aware Environment.* **In :** Proc. of European Symposium on Mobile Media Delivery (EuMob '06), sep. 2006, Alghero, Sardinia, Italy. pp 1–5.

[107] Reichl Peter, Egger Sebastian, Schatz Raimund, D'Alconzo Alessandro. *The Logarithmic Nature of QoE and the Role of the Weber-Fechner Law in QoE Assessment.* **In :** Proc. of the 2010 IEEE International Conference on Communications (ICC '10), may 2010, Cape Town, South Africa. IEEE Computer Society, 2010, pp 1–5.

[108] Ren Wei, Singh Sameer, Singh Maneesha, Zhu Yuesheng. *State-of-the-Art on Spatio-temporal Information-based Video Retrieval.* Journal of Pattern Recognition, 2009, vol. 42, n°2, pp. 267-282.

[109] Rensink Ronald A., O'Regan J. Kevin, Clark James J. *To see or not to See : The Need for Attention to Perceive Changes in Scenes.* Journal of Psychological Science, 1997, vol. 8, n°5, pp. 368-373.

[110] Rijsselbergen Dieter, Poppe Chris, Verwaest Maarten, Mannens Erik, Walle Rik. *Semantic Mastering : Content Adaptation in the Creative Drama Production Workflow.* Journal of Multimedia Tools and Applications, 2012, vol. 59, n°1, pp. 307-340.

[111] Russell Stuart Jonathan, Norvig Peter. *Artificial Intelligence : A Modern Approach (3rd Edition).* Prentice Hall, 3 edition, dec. 2009.

[112] School Cornell University Law. *Berne Convention for the Protection of Literary and Artistic Works(Paris Text 1971)* **[online]**. Available : <http://www.law.cornell.edu/treaties/berne/overview.html> (accessed 24.01.2013).

[113] Schwarz Heiko, Marpe Detlev, Wiegand Thomas. *Overview of the Scalable Video Coding Extension of the H.264/AVC Standard.* IEEE Transactions on Circuits and Systems for Video Technology, Sept. 2007, vol. 17, n°9, pp. 1103-1120.

[114] Shi Yonggang, Karl W. Clem. *Real-Time Tracking Using Level Sets.* **In :** Proc. of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 2, jun. 2005, Washington, DC, USA. IEEE Computer Society, 2005, pp 34–41.

[115] Shih T.K., Chang Rong-Chi. *Digital Inpainting - Survey and Multilayer Image Inpainting Algorithms.* **In :** Proc. of the 3^{rd} International Conference in Information Technology and Applications (ICITA '05), volume 1, jul. 2005, Taipei, Taiwan. IEEE Computer Society, 2005, pp 15–24.

[116] Sidiropoulos P. *Differential Edit Distance : A Metric for Scene Segmentation Evaluation.* IEEE Transactions on Circuits and Systems for Video Technology, jun. 2012.

[117] Simons Daniel J., Levin Daniel T. *Change Blindness.* Journal of Trends in Cognitive Sciences, oct. 1997, vol. 1, n°7, pp. 261-267.

[118] Sinha Prabhakant, Zoltners Andris A. *The Multiple-Choice Knapsack Problem.* Journal of Operations Research, 1979, vol. 27, n°3, pp. 503-515.

[119] Smeaton A.F., Over P., Doherty A. R. *Video shot boundary detection : Seven years of TRECVID activity.* Computer Vision and Image Understanding, apr. 2010, vol. 114, n°4, pp. 411–418.

[120] Smeulders Arnold W. M., Worring Marcel, Santini Simone, Gupta Amarnath, Jain Ramesh. *Content-Based Image Retrieval at the End of the Early Years.*

IEEE Transactions on Pattern Analysis and Machine Intelligence, dec. 2000, vol. 22, n°12, pp. 1349-1380.

[121] Smith J. R., Lugeon B. *A Visual Annotation Tool for Multimedia Content Description.* **In :** Proc. of the SPIE 4210 Photonics East, Internet Multimedia Management Systems, volume 4210, nov. 2000, Boston, MA, USA. SPIE,20, pp 49–59.

[122] Snoek Cees G. M., Worring Marcel. *Multimodal Video Indexing : A Review of the State-of-the-Art.* Journal of Multimedia Tools and Applications, 2003, vol. 25, n°1, pp. 5-35.

[123] Sofokleous Anastasis A, Angelides Marios C. *DCAF : an MPEG-21 Dynamic Content Adaptation Framework.* Journal of Multimedia Tools and Applications, 2008, vol. 40, n°2, pp. 151-182.

[124] Stegmaier Florian, Doeller Mario, Coquil David, El-Khoury Vanessa, Kosch Harald. *VAnalyzer : A MPEG-7 based Semantic Video Annotation Tool.* **In :** Proc. of the 11^{th} International Workshop of the Multimedia Metadata Community, volume 583, may 2010, Barcelona, Spain.

[125] Sun Kai, Yu Junqing, Huang Yue, Hu Xiaoqiang. *An Improved Valence-arousal Emotion Space for Video Affective Content Representation and Recognition.* **In :** Proc. of the IEEE International Conference on Multimedia and Expo (ICME '09), jul. 2009, New York, USA. IEEE Computer Society, 2009, pp 566–569.

[126] Thang TC, Jung YJ, Ro YM. *Modality Conversion for QoS Management in Universal Multimedia Access.* **In :** Proc. of IEE-Vision, Image and Signal Processing, volume 152. jun. 2005, IET, 2005, pp 374–384.

[127] Timmerer Christian. *Generic Adaptation of Scalable Multimedia Resources.* VDM Verlag.

[128] Tjondronegoro Dian. *Content-based Video Indexing for Sports Applications using Integrated Multi-Modal Approach.* Phd thesis. Melbourne : University of Deakin, 2005, 320 p.

[129] University Mannheim. *MoCA Project : Automatic Scaling and Cropping Videos* **[online]**. Available : <http://pi4.informatik.uni-mannheim.de/pi4.data/content/projects/moca/Project-ResolutionAdaptation.html> (accessed 16.01.2013).

[130] Valdés Víctor, Martínez José M. *Content Adaptation Tools in the CAIN Framework.* **In :** Visual Content Processing and Representation. Springer Berlin Heidelberg, sep. 2006, volume 3893, pp 9–15.

[131] van Beek Peter, Smith John R., Ebrahimi Touradj, Suzuki Teruhiko, Askelof Joel. *Metadata-Driven Multimedia Access.* IEEE Signal Processing Magazine, mar. 2003, vol. 20, n°2, pp. 40-52.

[132] Van Deursen Davy, De Neve Wesley, Van Lancker Wim, Van de Walle Rik. *Semantic adaptation of synchronized multimedia streams in a format-independent way.* **In :** Picture Coding Symposium (PCS' 09), may 2009, Chicago, IL, USA. IEEE Computer Society, 2009, pp 17–20.

[133] Van Deursen Davy, Van Lancker Wim, De Neve Wesley, Paridaens Tom, Mannens Erik, Van de Walle Rik. *NinSuna : a Fully Integrated Platform for Format-Independent Multimedia Content Adaptation and Delivery using Semantic Web Technologies.* Journal of Multimedia Tools and Applications, 2010, vol. 46, n°2-3, pp. 371-398.

[134] Van Deursen Davy, Van Lancker Wim, Paridaens Tom, De Neve Wesley, Mannens Erik, Van de Walle Rik. *NinSuna : A Format-Independent Multimedia Content*

Adaptation Platform Based on Semantic Web Technologies. **In :** Proceedings of the 10^{th} IEEE International Symposium on Multimedia, ISM '08, 2008, Washington, DC, USA. IEEE Computer Society, 2008, pp 491–492.

[135] van Ossenbruggen Jacco, Stamou Giorgos, Pan Jeff Z. *Multimedia Annotations on the Semantic Web.* IEEE MultiMedia magazine, 2005, vol. 13, n°1, pp. 86-90.

[136] Vetro Anthony, Christopoulos Charilaos A., Ebrahimi Touradj. *Special Issue on Universal Multimedia Access.* IEEE Signal Processing Magazine, 2003, vol. 20, n°2, pp. 16.

[137] Vezhnevets Vladimir. *"GrowCut" - Interactive Multi-Label N-D Image Segmentation By Cellular Automata.* Cybernetics, 2004, vol. 127, n°2, pp. 150-156.

[138] Wang Yao, Kim Jae-Gon, Chang Shih-Fu., Kim Hyung-Myung. *Utility-Based Video Adaptation for Universal Multimedia Access (UMA) and Content-Based Utility Function Prediction for Real-Time Video Transcoding.* IEEE Transactions on Multimedia, feb. 2007, vol. 9, n°2, pp. 213-220.

[139] Wang Yao, Liu Zhu, Huang Jin-Cheng. *Multimedia Content Analysis-Using Both Audio and Visual Clues.* IEEE Signal Processing Magazine, nov. 2000, vol. 17, n°6, pp. 12-36.

[140] Wang Zhou, Bovik Alan C. *Mean Squared Error : Love it or leave it ? A new look at Signal Fidelity Measures.* IEEE Signal Processing Magazine, jan. 2009, vol. 26, n°1, pp. 98-117.

[141] Wang Zhou, Sheikh Hamid R., Bovik Alan C. *Objective Video Quality Assessment.* **In :** Handbook of Video Databases : Design and Applications (Internet and Communications). 1^{st} edition, USA, CRC Press, sep. 2003, pp 1041–1078.

[142] Weber Ernst Heinrich. *De Pulsu, Resorptione, Auditu Et Tactu. Annotationes Anatomicae Et Physiologicae.* Lipsiae : prostat apud C.F. Koehler, 1834, 248 p.

[143] Webster Arthur A, Jones Coleen T, Pinson Margaret H, Voran Stephen D, Wolf Stephen. *Objective Video Quality Assessment System based on Human Perception.* **In :** Proc. of the SPIE. 1913, Human Vision, Visual Processing, and Digital Display IV, volume 1913, jan. 1993, San Jose, CA, USA. SPIE, 1993, pp 15–26.

[144] Wilson Benjamin, Cappelleri David, Simpson Timothy W, Frecker Mary. *Efficient Pareto frontier exploration using surrogate approximations.* Journal of Optimization and Engineering, 2001, vol. 2, n°1, pp. 31-50.

[145] Wu Wanmin, Arefin Ahsan, Rivas Raoul, Nahrstedt Klara, Sheppard Renata, Yang Zhenyu. *Quality of Experience in Distributed Interactive Multimedia Environments : Toward a Theoretical Framework.* **In :** Proc. of the 17^{th} ACM international conference on Multimedia (MM '09), oct. 2009, Beijing, China. ACM, 2009, pp 481–490.

[146] Yilmaz Alper, Javed Omar, Shah Mubarak. *Object Tracking : A Survey.* Journal of ACM Computing Surveys (CSUR), dec. 2006, vol. 38, n°4, pp. 45.

[147] Yilmaz Alper, Li Xin, Shah Mubarak. *Contour-Based Object Tracking with Occlusion Handling in Video Acquired using Mobile Cameras.* IEEE Transactions on Pattern Analysis and Machine Intelligence, nov. 2004, vol. 26, n°11, pp. 1531-1536.

[148] Zhu Song Chun, Yuille Alan. *Region Competition : Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, sep. 1996, vol. 18, n°9, pp. 884-900.

[149] Zufferey Michael, Kosch Harald. *Semantic Adaptation of Multimedia Content.* **In :** Proc. of the 48^{th} International Symposium ELMAR '06 focused on Multimedia Signal Processing and Communications, jun. 2006, Zadar, Croatia.

List of Publications

2013

International journals with reviewing committee

El-Khoury Vanessa, Jergler Martin, Getnet Abebe, Coquil David, Kosch Harald. *Fine granularity Semantic Video Annotation : an approach based on Automatic Shot level Concept Detection and Object Recognition.* International Journal of Pervasive Computing and Communications, Volume 9, Number 3, 2013, pp 243-269.

2012

International journals with reviewing committee

El-Khoury Vanessa, Coquil David, Bennani Nadia, Brunie Lionel. *Personalized vIdeo Adaptation Framework (PIAF) : High-Level Semantic Adaptation.* Multimedia Tools and Applications, 2012, pp 1-42.

International conferences with reviewing committee

El-Khoury Vanessa, Jergler Martin, Coquil David, Kosch Harald. *Semantic Video Content Annotation at the Object Level.* In : Proceedings of the 10^{th} International Conference on Advances in Mobile Computing & Multimedia, MoMM '12, 2012,New York, NY, USA. ACM, pp 179-188.

2010

International conferences with reviewing committee

El-Khoury Vanessa, Bennani Nadia, Coquil David. *Utility Function for Semantic Video Content Adaptation.* In : ACM Masters and Doctoral Colloquium(MDC) in conjunction with IIWAS2010. December 2010, ACM, pp 921-924.

Demo

Stegmaier Florian, Doeller Mario, Coquil David, El-Khoury Vanessa, Kosch Harald. *VAnalyzer : A MPEG-7 based Semantic Video Annotation Tool.* Workshop on Interoperable Social Multimedia Applications (WISMA), 2010.

2009

International conferences with reviewing committee

El-Khoury Vanessa. *A Multi-level Access Control Scheme for Multimedia Database.* In : Proceedings of the 9^{th} Workshop on Multimedia Metadata (WMM'09), Toulouse(France). 2009.

El-Khoury Vanessa, Bennani Nadia, Ouksel Aris M. *Distributed Key Management in Dynamic Outsourced Databases : A Trie-Based Approach.* In :

Proceedings of the 2009 First International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA '09, 2009, Washington, DC, USA. IEEE Computer Society, pp 56-61.

Annex

Annex A

Fundamental Properties of Pixel (adopted from [36])

This annex is included for informational purposes only. It recalls some fundamental properties of the pixel adopted from [36], which we need it in the formal definition of the region (see Definition 5).

Neighborhoods of a Pixel : Let a pixel p of coordinates (i, j) in the frame f. The set of pixels given by : $N_4(p) = \{ (i+1, j), (i-1, j), (i, j+1), (i, j-1) \})$ is called its $4-neighbors$, such as each of these neighbors is at one-unit distance from p. Its $8-neighbors$ given by $N_8(p) = N_4(p) \cup \{ (i+1, j+1), (i+1, j-1), (i-1, j+1), (i-1, j-1) \})$ contains its $4-neighbors$ pixels and its diagonal neighbors $(x \pm 1, y \pm 1)$. If the pixel is a on the border of the image, thus its $4-neighbors$ and $8-neighbors$ are subsets of $N_4(p)$ and $N_8(p)$, respectively.

Property 5. : If a pixel q is a $4-neighbor$ of p then q is also a $8-neighbor$ of p. This property is a direct implication of the definition of the $8-neighbors$.

Figure A.1 illustrates the neighborhoods of the pixel of coordinates (i, j), such that the green cells are its $4-neighbors$ and the blue cells are its diagonal neighbors. The set of blue and green cells are its $8-neighbors$

(i - 1, j - 1)	(i - 1, j)	(i - 1, j + 1)
(i, j - 1)	(i, j)	(i, j + 1)
(i + 1, j - 1)	(i + 1, j)	(i + 1, j + 1)

FIGURE A.1 – Neighborhoods of pixels with the coordinates (i, j).

$4-(8)$ *connectivity* **:** Generally, the connectivity is based on $4-$, $8-$ or $m-$ *connectivity*. In this thesis, we just use the $4-$ or $8-$ *connectivity*. Two pixels p and q in a frame f having the same characteristic $phm_c q$ are :

- $4-$ *connected* if q is element of the set $N_4(p)$ and,
- $8-$ *connected* if q is element of the $N_8(p)$.

Pixel Connectivity : A pixel p is connected to a pixel q if p is $4-(8)$ *connected* to q or in a recursive way, if p is $4-(8)$ *connected* to a third pixel which itself is connected to q. In other words, two pixels q and p are connected if there is a path from p to q on which each pixel is $4-(8)$ *connected* to the next one.

$4-(8)$ *adjacency* **:** Two pixels p and q are $4-(8)$ *adjacent* if they are $4-(8)$ *connected*.

$4-(8)$ adjacency of Pixels Sets : Given two disjoint sets of pixels in the frame f, S_i^f and S_j^f, S_i^f is $4-(8)$*adjacent* to S_j^f if : $\exists\ p \in S_i^f$ and $\exists\ q \in S_j^f$ such that p and q are $4-(8)$ *adjacent*.

Example 18. Figure A.2 illustrates two sets of pixels represented by two different colors. On the left, the sets are $4-adjacent$ and thus also $8-adjacent$. On the right, the sets are only $8-adjacent$.

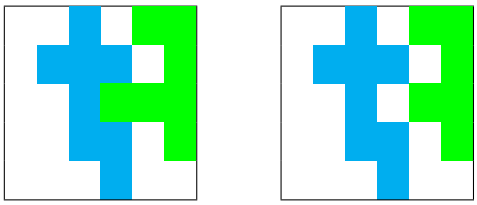


FIGURE A.2 – Illustration of the $4-(8)$ *adjacency* concept on the left and only $8-$ *adjacency* on the right.

Connected Component : Let $S^f \subseteq P^f$ be a set of pixels in a frame f such that all the pixels are connected. Then, the set S^f is said to be a connected component.

Example 19. Let us assume a binary frame such as the value 0 is used to refer to the background and the value 1 to the foreground. Figure A.3 illustrates two connected components based on $4-connectivity$. If the connectivity were based on $8-neighbors$, the two connected components would be merged into a single one.

0	0	1	0	1	1
0	1	1	1	0	1
0	0	1	0	1	1
0	0	1	1	0	1
0	0	0	1	0	0

FIGURE A.3 – A binary frame with two connected components based on $4-$ *connectivity*.

Annex B

Illustration of the Video Annotation Steps using SVCAT

This Annex comprises an example of annotating a video using SVCAT. It illustrates the steps of the annotation process described in Chapter 11, and provides excerpts from the Classification Schemes (CS) descriptions used in the example, as well as the output descriptions generated by the tool.

B.1 Input of SVCAT

SVCAT takes a video and two classification schemes as an input : priority and object terms classification schema. Figure B.1 illustrates an excerpt of images from the video used in our example. The video is extracted from the film 'Sex and the City'. As it can be observed from the figure, the video contains a lot of products placement. Assume that we want to annotate all the products placement using SVCAT. Thus, we need to upload a classification scheme, which consists of terms related to the products placement domain. An exemplary instance for the products placement classification schema is depicted in Listing 1, and the one for the priority classification scheme in Listing 2.

FIGURE B.1 – Excerpt of images from the video to be annotated by SVCAT

Listing 1 – Products Placement CS

```
<?xml version="1.0" encoding="UTF-8"?>
  <mpeg7:Mpeg7 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
               xsi:schemaLocation="urn:mpeg:mpeg7:schema:2004 mpeg7-v2.xsd">
    <mpeg7:Description xsi:type="mpeg7:ClassificationSchemeDescriptionType">
      <mpeg7:ClassificationScheme uri="urn:ProductPlacementCS">
        <mpeg7:Term termId="1">
          <mpeg7:Name xml:lang="en">Computer</mpeg7:Name>
          <mpeg7:Definition xml:lang="en">Product Placement for a computer</mpeg7:Definition>
          <mpeg7:Term termId="1.1">
            <mpeg7:Name xml:lang="de">HP-Computer</mpeg7:Name>
            <mpeg7:Definition xml:lang="en">Product Placement for an HP-Computer</mpeg7:Definition>
          </mpeg7:Term>
          <mpeg7:Term termId="1.2">
            <mpeg7:Name xml:lang="en">Apple-Computer</mpeg7:Name>
            <mpeg7:Definition xml:lang="en">Product Placement for an Apple-Computer</mpeg7:Definition>
          </mpeg7:Term>
        </mpeg7:Term>
        <mpeg7:Term termId="2">
          <mpeg7:Name xml:lang="en">Alcohol</mpeg7:Name>
          <mpeg7:Definition xml:lang="en">Product Placement for Alcohol</mpeg7:Definition>
          <mpeg7:Term termId="2.1">
            <mpeg7:Name xml:lang="en">Vodka</mpeg7:Name>
            <mpeg7:Definition xml:lang="en">Product Placement for Vodka</mpeg7:Definition>
          </mpeg7:Term>
        </mpeg7:Term>
        <mpeg7:Term termId="3">
      <mpeg7:Name xml:lang="en">Beer</mpeg7:Name>
          <mpeg7:Definition xml:lang="en">Product Placement for Beer</mpeg7:Definition>
          <mpeg7:Term termId="3.1">
            <mpeg7:Name xml:lang="en">Budweiser</mpeg7:Name>
            <mpeg7:Definition xml:lang="en">Product Placement for Budweiser</mpeg7:Definition>
          </mpeg7:Term>
        </mpeg7:Term>
        <mpeg7:Term termId="4"> ... </mpeg7:Term>
      </mpeg7:ClassificationScheme>
    </mpeg7:Description>
  </mpeg7:Mpeg7>
```

Listing 2 – Priority CS.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2    <mpeg7:Mpeg7 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3                 xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
4                 xsi:schemaLocation="urn:mpeg:mpeg7:schema:2004 mpeg7-v2.xsd"
                    >
5       <mpeg7:Description xsi:type="
            mpeg7:ClassificationSchemeDescriptionType">
6       <mpeg7:ClassificationScheme uri="urn:PriorityCS">
7          <mpeg7:Term termId="NoPriority">
8       <mpeg7:Name xml:lang="en">0.00</mpeg7:Name>
9       <mpeg7:Definition xml:lang="en">No</mpeg7:Definition>
10         </mpeg7:Term>
11         <mpeg7:Term termId="TopPriority">
12      <mpeg7:Name xml:lang="en">1</mpeg7:Name>
13      <mpeg7:Definition xml:lang="en">Top</mpeg7:Definition>
14         </mpeg7:Term>
15       </mpeg7:ClassificationScheme>
16       </mpeg7:Description>
17   </mpeg7:Mpeg7>
```

B.2 Temporal Structure Localization and Annotation

As mentioned in Chapter 11, SVCAT benefits from VAnalyzer, which performs an automatic detection of the shot boundaries [124]. Once the shots are detected, an interface representing an overview of the result of the detection algorithm appears. For instance, as can be observed in Figure B.2, the video is composed of 10 shots. Moreover, this interface implements the scene construction functionalities as explained in Section 11.3.1.

Once the scenes are constructed, a temporal structure description in MPEG-7 is generated by clicking the button 'Generate XML'. For instance, Listing 3 shows an excerpt of the generated MPEG-7 description from the video used in this example. Scenes and shots are represented by *VideoSegments*, which are hierarchically structured by nested *TemporalDecompositions*. The outer one represents the decomposition of the video into scenes (line 5) and each of the inner ones represents the decomposition of the scene into its shots (line 10).

Listing 3 – Annotation of temporal structure

```
1  ...
2     <ns1:Video>
3        <ns1:MediaInformation> ...</ns1:MediaInformation>
4        <ns1:CreationInformation> ... </ns1:CreationInformation>
5        <ns1:TemporalDecomposition>
6           <!-- Description of the first scene -->
7           <ns1:VideoSegment id="Scene_1">
8              <ns1:MediaTime> ... </ns1:MediaTime>
9              <!-- Description of the decomposition of the shot -->
10             <ns1:TemporalDecomposition>
11                <ns1:VideoSegment id="ID_0"> ... </ns1:VideoSegment>
12             </ns1:TemporalDecomposition>
13          </ns1:VideoSegment>
14       </ns1:TemporalDecomposition>
15    ...
```

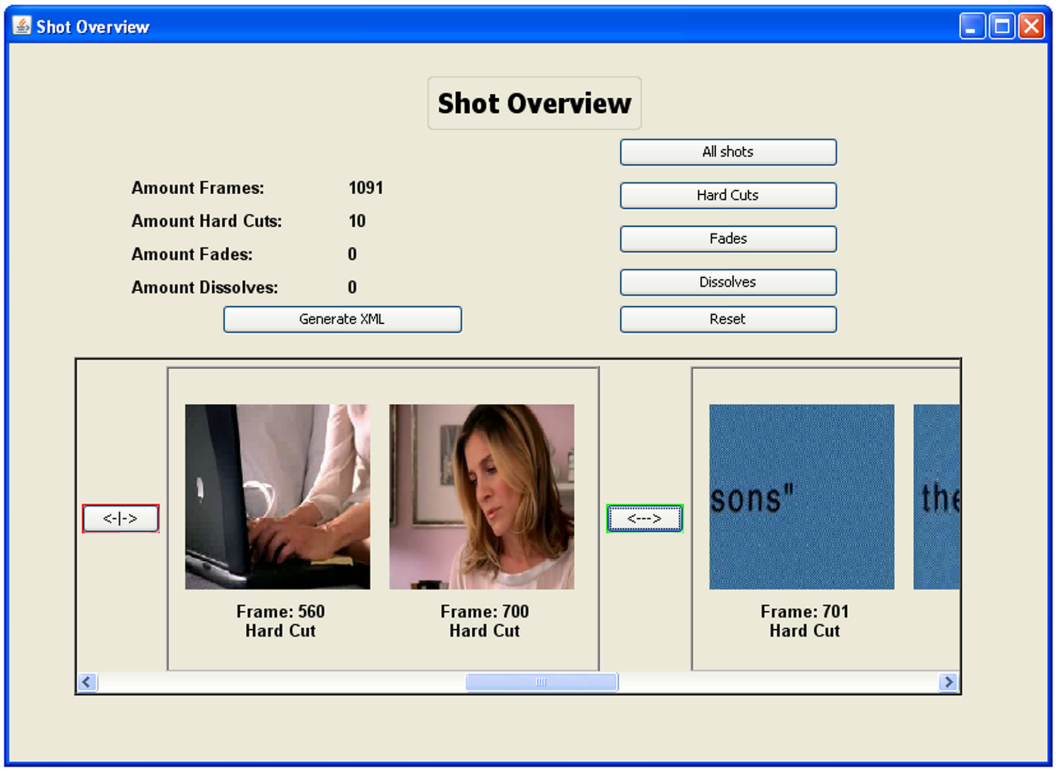


FIGURE B.2 – Overview of the shots detection in VAnalyzer

```
16      </ns1:Video>
17  ...
```

Based on this generated metadata, the temporal structure of the video is displayed to the annotator via the scene panel of SVCAT. Figure B.3) shows the 3^{rd} scene of the video in the top part, and displays its shots organization, in the bottom part. Moreover, a top priority is assigned to all the shots, which corresponds to the value 1 according to Listing 2).

B.3 Spatio-temporal Structure Localization and Annotation

In the remainder of this section, we describe in detail the representation of the object metadata. In our approach, we decouple semantic metadata from the structural metadata in order to achieve a less verbose annotation. Thus, the object description is split into two parts : *static* and *dynamic* part.

The *static* part corresponds to a concrete instance of an object along with its semantics. This annotation is created when the user selects an object and attaches a descriptive term to it. The object is annotated using a *MovingRegion* descriptor, and linked to a descriptive term of the CS. Listing 4 shows an example of a static object description. The object is annotated using a *MovingRegion* descriptor, identified by the *id* attribute and linked to a descriptive term in the CS using the *href* attribute . Note that the *id* is a concatenation of 'SvcatMR', 'timestamp' and the 'term' describing the object semantic.

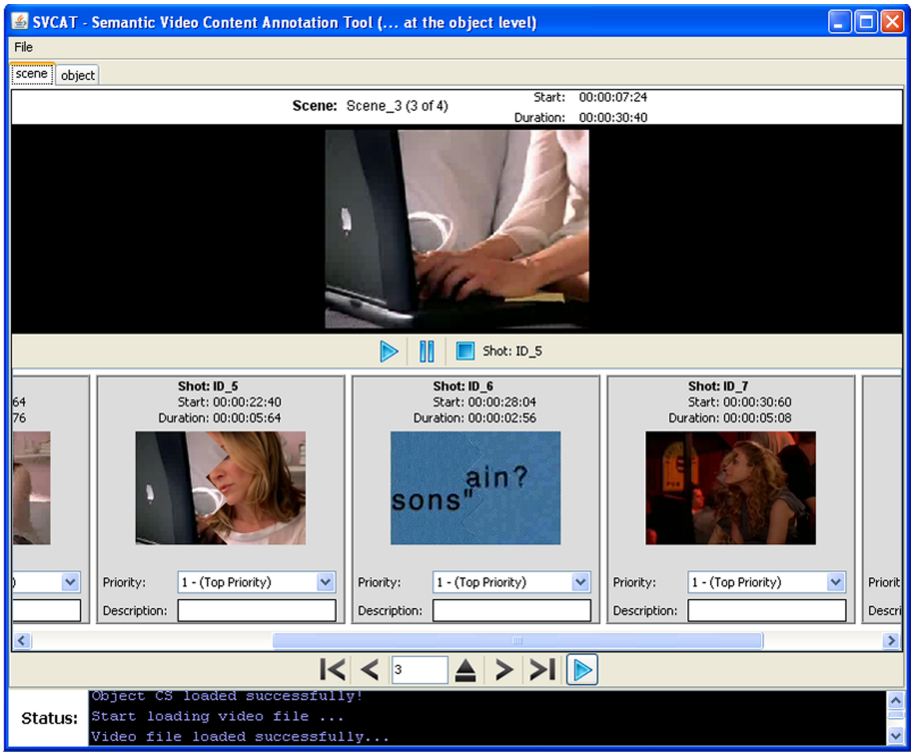


FIGURE B.3 – Scene panel of SVCAT

Listing 4 – Static object description

```
<ns1:SpatialDecomposition>
  <ns1:MovingRegion id="SvcatMR1330819412576Apple_Computer">
    <ns1:SemanticRef href="urn_ProductPlacementCS_1 . 2"/>
  </ns1:MovingRegion>
  <ns1:MovingRegion> ... </ns1:MovingRegion>
</ns1:SpatialDecomposition>
```

The *dynamic* part represents the information related to the spatial segmentation (i.e., the contour of the object and its size in a frame) and the temporal segmentation (i.e, its appearance with respect to scene and shot structure). To represent the spatial information at pixel accuracy, the usual MPEG-7 descriptors are not expressive enough. For instance, the MPEG-7 *RegionLocator* only allows the annotation of simple geometric shapes (at most polygons). A more expressive possibility would be the *SpatioTemporalLocator* in combination with a *FigureTrajectory*, but it lacks precision as well. Indeed, this representation of regions is based on parametric curves along with interpolation functions. These functions are expensive to calculate and the resulting curve only provides an approximation to the accurate contour. Thus, we opted for gathering the position information in a separate XML document according to the schema depicted in Listing 5.

Listing 5 – XML schema for position information

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified">
    <!-- Definition of ObjectInformation -->
    <xs:element name="ObjectInformation">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="TimeStamp" type="TimeStampType"
                    minOccurs="1" maxOccurs="unbounded"/>
            </xs:sequence>
```

```
9           </xs:complexType>
10      </xs:element>
11
12      <!-- Definition of TimeStampType -->
13      <xs:complexType name="TimeStampType">
14          <xs:sequence>
15              <xs:element name="FrameNumber" type="xs:integer" use="
                    required"/>
16              <xs:element name="ObjectSize" type="xs:integer" use="required
                    "/>
17              <xs:element name="ObjectContour" type="xs:string" use="
                    required"/>
18          </xs:sequence>
19          <xs:attribute name="time" type="xs:integer" use="required"/>
20      </xs:complexType>
21
22  </xs:schema>
```

The root element, *ObjectInformation* consists of an unbounded number of *TimeStamp* elements, where each one is a sequence of three elements *FrameNumber*, *ObjectSize* and *ObjectContour*, with an attribute *time*. The *time* attribute provides a temporal description, the position of the frame enclosing the object at a distinct point in time. In addition to the media time, we store the frame sequence number, the object size (i.e., the number of pixels that form the object in the frame) and, of course, the *ObjectContour*. The latter is obtained by transforming the frame containing the tracked object into its binary mask, with 0 and 1 values representing object pixels and background pixels, respectively. For matters of size and performance, we use run-length encoding to encode the result and store it as a string. The run length encoding scans the frame from left to right in first order and from top to bottom in second order. Note that the original mask can be easily re-established based on the resolution of the video. An example instance of this schema is given in Listing 8.

Besides spatial information, the *dynamic* part also describes the object frame sequences related to an object in a shot (refer to Section 7.3.3). An example of dynamic object description is depicted in listing 6. The excerpt consists of a *TemporalDecomposition*, which is embedded in the *VideoSegment* of the corresponding shot. Each sub-*VideoSegment* corresponds to an *SOFS*, which is identified by the *id* attribute. It references both the .xml document that holds the position information (line $4-8$) and, the static description part using the *MovingRegionRef* descriptor (line 19). In addition, the dynamic description contains a *TemporalMask* (line $9-16$), which describes the exact time interval in which the object occurs within the shot the first time.

Listing 6 – Dynamic object description

```
1  <!-- Description of the Shot Object Frame Sequences containing the Apple
       Logo Object -->
2  <ns1:TemporalDecomposition>
3    <ns1:VideoSegment id="SOFS_ID_0_1.2 - (Apple_Computer)">
4     <ns1:MediaLocator>
5      <ns1:MediaUri>
6        SvcatMR1330819412576Apple_ComputerID_5.xml
7      </ns1:MediaUri>
8     </ns1:MediaLocator>
9     <ns1:TemporalMask>
10     <ns1:SubInterval>
11       <ns1:MediaTimePoint>
12         2012-03-04T00:00:22:400F1000
```

```
      </ns1:MediaTimePoint>
      <ns1:MediaDuration>PT00H00M00S01N120F</ns1:MediaDuration>
    </ns1:SubInterval>
  </ns1:TemporalMask>
  <!-- Reference to static, semantic description-->
  <ns1:SpatioTemporalDecomposition>
    <ns1:MovingRegionRef href="SvcatMR1330819412576-Apple_Computer"/>
  </ns1:SpatioTemporalDecomposition>
 </ns1:VideoSegment>
</ns1:TemporalDecomposition>
```

B.4 Output of SVCAT

Listing 7 – Output annotation of the video

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns1:Mpeg7 xmlns:ns1="urn:mpeg:mpeg7:schema:2004">
  <ns1:DescriptionUnit xmlns:xsi="http://www.w3.org/2001/XMLSchema-
      instance" xsi:type="ns1:VideoType">
    <ns1:Video>
      <ns1:MediaInformation>
        <ns1:MediaIdentification> ... </ns1:MediaIdentification>
        <ns1:MediaProfile>
          <ns1:MediaFormat>
            <ns1:Content href="urn:mpeg:mpeg7:cs:ContentCS:2001:2"
                >
          <Name>Audiovisual</Name>
         </ns1:Content>
            <ns1:FileFormat href="
                urn:mpeg:mpeg7:cs:FileFormatCS:2001:7"> ... </
                ns1:FileFormat>
            <ns1:FileSize>8897564</ns1:FileSize>
            <ns1:VisualCoding> ... </ns1:VisualCoding>
          </ns1:MediaFormat>
          <ns1:MediaInstance>
            <ns1:InstanceIdentifier/>
             <!-- Path of the video in use -->
            <ns1:MediaLocator>
              <ns1:MediaUri>file:/F:/VideoDatabase/Video.avi</
                  ns1:MediaUri>
            </ns1:MediaLocator>
          </ns1:MediaInstance>
        </ns1:MediaProfile>
      </ns1:MediaInformation>
      <ns1:CreationInformation>
        <ns1:Creation>
          <ns1:Title>Video.avi</ns1:Title>
          <ns1:CreationTool>
            <ns1:Tool>
              <ns1:Definition>Vanalyzer</ns1:Definition>
            </ns1:Tool>
          </ns1:CreationTool>
        </ns1:Creation>
      </ns1:CreationInformation>
      <ns1:TemporalDecomposition>
        <!-- Description of the first scene -->
        <ns1:VideoSegment id="Scene_1">
          <ns1:MediaTime>
            <ns1:MediaTimePoint>00:00:00:000</ns1:MediaTimePoint>
```

```
40                    <ns1:MediaDuration>PT00H00M00S04N920F</
                          ns1:MediaDuration>
41                 </ns1:MediaTime>
42                 <!-- Description of the decomposition of the shot -->
43                 <ns1:TemporalDecomposition>
44                    <ns1:VideoSegment id="ID_0">
45                       <ns1:TextAnnotation>
46                          <ns1:FreeTextAnnotation>Watching</
                                ns1:FreeTextAnnotation>
47                       </ns1:TextAnnotation>
48                       <ns1:MediaTime>
49                          <ns1:MediaTimePoint>00:00:00:000</
                                ns1:MediaTimePoint>
50                          <ns1:MediaDuration>PT00H00M00S04N920F</
                                ns1:MediaDuration>
51                       </ns1:MediaTime>
52                       <ns1:SpatioTemporalDecomposition overlap="false"
                             gap="false">
53                          <ns1:StillRegion>
54                             <ns1:CreationInformation>
55                                <ns1:Creation>
56                                   <ns1:Title type="Frame 0"></ns1:Title
                                         >
57                                </ns1:Creation>
58                             </ns1:CreationInformation>
59                             <ns1:TextAnnotation>
60                                <ns1:KeywordAnnotation>
61                                   <ns1:Keyword>Hard Cut</ns1:Keyword>
62                                </ns1:KeywordAnnotation>
63                             </ns1:TextAnnotation>
64                             <ns1:MediaTimePoint>00:00:00:000</
                                   ns1:MediaTimePoint>
65                          </ns1:StillRegion>
66                          <ns1:StillRegion>
67                             <ns1:CreationInformation>
68                                <ns1:Creation>
69                                   <ns1:Title type="Frame 122"></
                                         ns1:Title>
70                                </ns1:Creation>
71                             </ns1:CreationInformation>
72                             <ns1:TextAnnotation>
73                                <ns1:KeywordAnnotation>
74                                   <ns1:Keyword>Hard Cut</ns1:Keyword>
75                                </ns1:KeywordAnnotation>
76                             </ns1:TextAnnotation>
77                             <ns1:MediaTimePoint>00:00:04:880</
                                   ns1:MediaTimePoint>
78                          </ns1:StillRegion>
79                       </ns1:SpatioTemporalDecomposition>
80                    </ns1:VideoSegment>
81                 </ns1:TemporalDecomposition>
82              </ns1:VideoSegment>
83              <!-- Description of the second scene -->
84              <ns1:VideoSegment id="Scene_2"> ... </ns1:VideoSegment>
85              <!-- Description of the third scene -->
86              <ns1:VideoSegment id="Scene_3">
87                 <ns1:SemanticRef href="urn_PriorityScaleCS_TopPriority"/>
88                 <ns1:MediaTime>
89                    <ns1:MediaTimePoint>2012-03-04T00:00:07:240F1000</
                          ns1:MediaTimePoint>
90                    <ns1:MediaDuration>PT00H00M00S36N440F</
                          ns1:MediaDuration>
```

```
91                </ns1:MediaTime>
92                <ns1:TemporalDecomposition>
93                   <ns1:VideoSegment id="ID_3"> ... </ns1:VideoSegment>
94                   <ns1:VideoSegment id="ID_4"> ... </ns1:VideoSegment>
95                   <ns1:VideoSegment id="ID_5">
96                      <!-- Description of the priority of the fifth shot
                           -->
97                      <ns1:SemanticRef href="
                           urn_PriorityScaleCS_TopPriority"/>
98                      <ns1:MediaTime>
99                         <ns1:MediaTimePoint>2012-03-04T00:00:22:400F1000
                              </ns1:MediaTimePoint>
100                        <ns1:MediaDuration>PT00H00M00S05N640F </
                              ns1:MediaDuration>
101                     </ns1:MediaTime>
102                  <ns1:SpatioTemporalDecomposition gap="false" overlap="
                        false"> ...
103                  </ns1:SpatioTemporalDecomposition>
104                  <!-- Description of the Shot Object Frame Sequences
                        containing the Apple_Computer Object -->
105                  <ns1:TemporalDecomposition>
106                     <ns1:VideoSegment id="SOFS_ID_5_1.2 - (
                           Apple_Computer)">
107                        <ns1:MediaLocator>
108                           <ns1:MediaUri>
                                 SvcatMR1330819412576Apple_Computer\-
109                              SvcatMR1330819412576Apple_ComputerID_5.xml
110                           </ns1:MediaUri>
111                        </ns1:MediaLocator>
112                        <ns1:TemporalMask>
113                           <ns1:SubInterval>
114                              <ns1:MediaTimePoint>2012-03-04
                                    T00:00:22:400F1000
115                              </ns1:MediaTimePoint>
116                              <ns1:MediaDuration>PT00H00M00S01N120F
117                              </ns1:MediaDuration>
118                           </ns1:SubInterval>
119                        </ns1:TemporalMask>
120                         <!-- Description of the semantic and priority of
                                 the object -->
121                        <ns1:SpatioTemporalDecomposition>
122                           <ns1:MovingRegionRef href="
                                 SvcatMR1330819412576-Apple_Computer"/>
123                           <ns1:SemanticRef href="
                                 urn_PriorityScaleCS_TopPriority"/>
124                        </ns1:SpatioTemporalDecomposition>
125                     </ns1:VideoSegment>
126                  </ns1:TemporalDecomposition>
127                     </ns1:VideoSegment>
128                     <ns1:VideoSegment id="ID_6"> ... </ns1:VideoSegment>
129                     <ns1:VideoSegment id="ID_7"> ... </ns1:VideoSegment>
130                     <ns1:VideoSegment id="ID_8"> ... </ns1:VideoSegment>
131                  </ns1:TemporalDecomposition>
132                  </ns1:VideoSegment>
133                  <!-- Description of the fourth scene -->
134                  <ns1:VideoSegment id="Scene_4"> ...
135                     <ns1:VideoSegment id="ID_9"> ... </ns1:VideoSegment>
136                  </ns1:VideoSegment>
137            </ns1:TemporalDecomposition>
138            <!-- Semantic Salient Objects Description -->
139            <ns1:SpatialDecomposition>
140               <!-- Description related to the Apple Logo -->
```

```
141             <ns1:MovingRegion id="SvcatMR1330819412576Apple_Computer">
142                 <ns1:SemanticRef href="urn_ProductPlacementCS_1.2"/>
143             </ns1:MovingRegion>
144             <ns1:MovingRegion ... </ns1:MovingRegion>
145         </ns1:SpatialDecomposition>
146     </ns1:Video>
147 </ns1:DescriptionUnit>
148 <ns1:Description xsi:type="ns1:SemanticDescriptionType">
149     <ns1:Semantics>
150         <ns1:SemanticBase xsi:type="ns1:SemanticType"
151         id="urn_PriorityScaleCS_TopPriority">
152             <ns1:Label href="urn:PriorityScaleCS:TopPriority"/>
153         </ns1:SemanticBase>
154         <ns1:SemanticBase xsi:type="ns1:SemanticType" id="
            urn_ProductPlacementCS_1.2">
155             <ns1:Label href="urn:ProductPlacementCS:1.2"/>
156         </ns1:SemanticBase>
157     </ns1:Semantics>
158 </ns1:Description>
159 </<ns1:Mpeg7>
```

Listing 8 shows an example instance of the spatio-temporal description for the apple product placement, which appears in the 5^{th} shot as depicted in Listing 6 at line 119. The sample sketches the position description for the object occurrence in the first two frames of the 5^{th} shot. The first frame is described in lines $4-10$ and the second frame in lines $12-16$. Note that time information is given in nanoseconds. Moreover, the run-length encoded string in line 8 has the following meaning : 30101 occurrences of 1's , followed by 3 occurrences of 0 and so on; whereas value 1 means that the pixel belongs to the object and 0 otherwise.

Listing 8 – Annotation of the spatio-temporal structure of the object

```
1  <!--Object Information Description within an SOFS (
       SvcatMR1330819412576Apple_ComputerID_5.xml) -->
2
3  <ObjectInformation>
4      <TimeStamp time="22400000000">
5          <FrameNumber>560</FrameNumber>
6          <ObjectSize>309</ObjectSize>
7          <ObjectContour>
8              (30101,1)(3,0)(315,1)(8,0) ...
9          </ObjectContour>
10     </TimeStamp>
11     <TimeStamp time="22440000000"> ... </TimeStamp>
12     <TimeStamp time="23360000000"> ... </TimeStamp>
13         <FrameNumber>561</FrameNumber>
14             <ObjectSize>289</ObjectSize>
15             <ObjectContour> ... </ObjectContour
16     </TimeStamp>
17     ...
18 </ObjectInformation>
```

Annex C

Example of an MPEG Profile Description

Listing 9 – Description of an MPEG user profile

```
<?xml version="1.0" encoding="UTF-8"?>
<dia:DIA xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-DIA-NS UED-2nd.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS" xmlns:mpeg7="
    urn:mpeg:mpeg7:schema:2004">
    <dia:Description xsi:type="dia:UsersType">
        <dia:User>
            <!-- Description of the user personal information -->
            <dia:UserCharacteristic xsi:type="dia:UserInfoType">
                <dia:UserInfo xsi:type="mpeg7:PersonType">
                    <mpeg7:Name>
                        <mpeg7:GivenName>Pascal</mpeg7:GivenName>
                        <mpeg7:FamilyName>Doe</mpeg7:FamilyName>
                    </mpeg7:Name>
                </dia:UserInfo>
            </dia:UserCharacteristic>

            <!-- Description of the user preferences -->
            <dia:UserCharacteristic xsi:type="dia:UsagePreferencesType">
                <dia:UsagePreferences>
                    <mpeg7:UsageSemanticPreferences>
            <mpeg7:SemanticConstraint SmCt_ID=" ID_1">
                        <mpeg7:SemanticConstraintName>
                            ProductPlacement-Omitted</
                            mpeg7:SemanticConstraintName>
                        <mpeg7:Description>Remove product placement
                            including logo, text, object and video
                            segments</mpeg7:Description>
                        <mpeg7:KeywordList>
                            <!-- Referring to the termID in the
                                Classification Schema
                                ProductPlacementCS -->
                            <mpeg7:Keyword href="
                                urn:ProductPlacementCS:1.2">
                                <mpeg7:Name>Apple-Computer</
                                    mpeg7:Name>
                            </mpeg7:Keyword>
                             <mpeg7:Keyword href="
                                 urn:ProductPlacementCS:2.1">
                                <mpeg7:Name>Vodka</mpeg7:Name>
                            </mpeg7:Keyword>
                            <mpeg7:Keyword href="
                                urn:ProductPlacementCS:3.1">
```

```
32                                          <mpeg7:Name>Budweiser</mpeg7:Name>
33                                      </mpeg7:Keyword>
34                                  </mpeg7:KeywordList>
35                              </mpeg7:SemanticConstraint>
36                          </mpeg7:UsageSemanticPreferences>
37                      </dia:UsagePreferences>
38                  </dia:UserCharacteristic>
39                      </dia:UsagePreferences>
40                  </dia:UserCharacteristic>
41              </dia:User>
42      </dia:Description>
43  </dia:DIA>
```

Annex D

Syntax of the Instantiated Constraint

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault
       ="qualified" attributeFormDefault="unqualified">
3
4      <!-- Definition of InstantiatedConstraint -->
5      <xs:element name="InstantiatedConstraint">
6          <xs:complexType>
7              <xs:sequence>
8                  <xs:element name="Scene" type="SceneType" minOccurs="1"
                       maxOccurs="unbounded"/>
9              </xs:sequence>
10             <xs:attribute name="InstCt_ID" type="xs:string" use="required
                   "/>
11             <xs:attribute name="SmCt_REF" type="xs:string" use="required"
                   />
12             <xs:attribute name="Video_REF" type="xs:string" use="required
                   "/>
13         </xs:complexType>
14     </xs:element>
15
16     <!-- Definition of SceneType -->
17     <xs:complexType name="SceneType">
18         <xs:sequence>
19             <xs:element name="Shot" type="ShotType" minOccurs="1"
                   maxOccurs="unbounded"/>
20         </xs:sequence>
21   <xs:attribute name="Scene_ID" type="xs:string" use="required"/>
22   <xs:attribute name="Scene_SFrame" type="xs:string" use="required"/>
23   <xs:attribute name="Scene_Length" type="xs:string" use="required"/>
24     </xs:complexType>
25
26     <!-- Definition of ShotType -->
27     <xs:complexType name="ShotType">
28         <xs:sequence>
29       <xs:element name="SOFS" type="SOFSType" minOccurs="1" maxOccurs="1"
           />
30   </xs:sequence>
31   <xs:attribute name="Shot_ID" type="xs:string" use="required"/>
32   <xs:attribute name="Shot_SFrame" type="xs:string" use="required"/>
33   <xs:attribute name="Shot_Length" type="xs:string" use="required"/>
34   <xs:attribute name="Shot_Priority" type="xs:string" use="required"/>
35     </xs:complexType>
36
37     <!-- Definition of SOFSType-->
38     <xs:complexType name="SOFSType">
39         <xs:sequence>
```

```
40              <xs:element name="ofs" type="ofsType" minOccurs="1" maxOccurs
                    ="unbounded"/>>
41          </xs:sequence>
42          <xs:attribute name="SOFS_ID" type="xs:string" use="required"/>
43      </xs:complexType>
44
45      <!-- Definition of ofsType-->
46      <xs:complexType name="ofsType">
47          <xs:sequence>
48              <xs:element name="ObjFrame" type="ObjFrameType" use="required
                    "/>
49          </xs:sequence>
50          <xs:attribute name="ofs_ID" type="xs:string" use="required"/>
51    <xs:attribute name="ofs_SFrame" type="xs:string" use="required"/>
52    <xs:attribute name="ofs_Length" type="xs:string" use="required"/>
53      </xs:complexType>
54
55      <!-- Definition of ObjFrameType-->
56      <xs:complexType name="ObjFrameType">
57          <xs:sequence>
58              <xs:element name="ObjectSize" type="xs:integer" use="required
                    "/>
59       <xs:element name="ObjectPriority" type="xs:integer" use="required"/>
60              <xs:element name="ObjectContour" type="xs:string" use="
                    required"/>
61          </xs:sequence>
62          <xs:attribute name="ObjFrame_NB" type="xs:string" use="required"/
                >
63      </xs:complexType>
64
65  </xs:schema>
```

Annex E

Heuristic for the Adaptation Plan Computation

This Annex describes the heuristic that we implemented in the framework to solve the optimization problem, which appears in the case of conflicting constraints during the computation of an adaptation plan. This heuristic is adopted with a slight modification from the approach proposed by Pisinger in [101]. Furthermore, we evaluate the performance of the heuristic for finding the adaptation plan that satisfies both end-user and owner constraints, and has a global utility value close to the optimal plan. To begin with, we describe the simulations that we used to analyze the performance of our heuristic in terms of its running time with respect to the number of shots. Then, we proceed to evaluate the quality of the generated plans by comparing them to that of the optimal plans obtained by an exhaustive search algorithm, which remains possible when the number of shots is low.

E.1 Description of the Algorithm

The algorithm takes as input the matrices U and S for $|Adp\mathcal{SH}_{sc,o}|$ disjoint classes $C_1, \ldots, C_{|Adp\mathcal{SH}_{sc,o}|}$ of $|\mathcal{OP}'|$ items each, as well as the percentage extracted from the owner constraint, in order to calculate the $maxfnb$ value. Moreover, it initialises a logical matrix A with the same numbers of row and column as the matrix S, and its elements $a_{i,j} = 1$.

The algorithm starts by computing the best adaptation plan $bap(sc, o)$ as described in Section 10.2.1, and the size sum $Ssum$ of the chosen items. If the condition $Ssum \leq maxfnb$ is satisfied, an optimal adaptation plan is found. It is sent to the adaptation execution engine to adapt the video. Otherwise, the next steps are performed.

The second step aims to find the undominated items sub-class R_j for each C_j, $j = 1, \ldots, |Adp\mathcal{SH}_{sc,o}|$. In fact, an item is said to be dominated if an item in the same class exists with higher utility and a lesser number of dropped frames. This is achieved by applying the dominance condition defined by Sinha and Zoltners [118], and formulated as follows :

Proposition 1 : If $p,\ q \in C_j$ with $s_{p,j} < s_{q,j}$ and $u_{p,j} > u_{q,j}$, then the item q is dominated by the item p, and $a_{q,j} = 0$ in every optimal solution to the problem.

As a consequence, several unpromising items are eliminated from each class, yielding an update to the logical matrix A (i.e., set $a_{i,j} = 0$ for each dominated item).

In the third step, the algorithm improves the solution iteratively by local shifts from an item p of the best adaptation plan to another item in $R_j - \{a_{p,j}\}$. Here, only a specific set of shifts need to be considered. Indeed, in order to contribute to the satisfaction of the owner constraint, a shift must result in a plan dropping fewer frames than the previous one. In our case, the operators have a fixed properties with respect to the number of dropped frames : $Drop-Object$ does not cause loss of frames ; and as an $SOFS$ is always contained in a shot, therefore $Drop-SOFS$ causes less loss of frames than $Drop-Shot$ (see Figure 10.5). Thus, the only shifts that must be considered are : $Drop-Shot$ to $Drop-SOFS$, $Drop-Shot$ to $Drop-Object$, and $Drop-SOFS$ to $Drop-Object$. Moreover, the shift should be chosen so that the loss of utility is minimized and the gain in frames is maximized. Thus, for each possible shift from item p to q in class $R_j - \{a_{p,j}\}$, we calculate the utility loss $ul_j(p,q)$ and the frame gain $fg_j(p,q)$. Since the shots are of different size, we normalize the value of the utility loss by multiplying by the size of the shot :

$$ul_j(p,q) = (u_{p,j} - u_{q,j}) \times size_{sh_j} \text{ and } fg_j(p,q) = (s_{p,j} - s_{q,j}) \\ \text{such that } q < p \text{ and } j = 1, \ldots, |Adp\mathcal{SH}_{sc,o}| \tag{E.1}$$

Accordingly, we define the slope $\Upsilon_j(p,q)$, that is the utility-to-size ratio obtained by shifting from item q to p, as follows :

$$\Upsilon_j(p,q) = \frac{ul_j(p,q)}{fg_j(p,q)} \tag{E.2}$$

After the slopes of all possible shifts are computed and ordered in an increasing order, the algorithm begins iteratively to select the shift with lowest slope until the condition $Ssum \leq maxfnb$ is satisfied. At each shift from p to q, the element $a_{p,j}$ of the matrix A is updated to 0. Once the condition is fulfilled and the feasible adaptation plan is found, the matrix A is updated. Indeed, all the items that doesn't belongs to the found plan are set to 0. Finally, the plan is sent to the adaptation execution engine to adapt the video.

Example 20. Let us consider $Adp\mathcal{SH}_{sc_1,o_2} = \{sh_1, sh_4, sh_7, sh_8\}$, such that $sc_1 \in \mathcal{SC}$ and $o_2 \in \mathcal{O}$ is the object to be removed, with an owner constraint applying : "No more than 10% of the scene can be removed". With scene sc_1 having a length $length_{sc_1} = 2880$ frames, the number of dropped frames should not exceed $maxfnb = 2880 * 0,1 = 288$. Figure E.1 illustrates the steps of the heuristic in order to select the best adaptation plan $bap(sc_1, o_2)$.

Step 1 : Find the best adaptation plan bap(sc_1, o_2).

Matrix U	sh_1	sh_4	sh_7	sh_8
op_1	0,50	0,80	0,75	0,34
op_2	0,65	1,00	0,90	0,67
op_3	0,70	0,40	0,20	0,80

$$\begin{array}{c} \textbf{Matrix S} \\ \textbf{op}_1 \\ \textbf{op}_2 \\ \textbf{op}_3 \end{array} \begin{array}{c} \begin{array}{cccc} \textbf{sh}_1 & \textbf{sh}_4 & \textbf{sh}_7 & \textbf{sh}_8 \end{array} \\ \left[\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 68 & \boxed{40} & \boxed{90} & 20 \\ \boxed{148} & 155 & 160 & \boxed{180} \end{array}\right] \end{array}$$

$\mathbf{bap(sc_1, o_2)} = (\ (op_3, sh_1),\ (op_2, sh_4),\ (op_2, sh_7),\ (op_3, sh_8)\ )$ *such that* :
$Ssum = 458 > maxfnb = 288 \implies$ Go to **Step 2**.

Step 2 : Find the undominated class R_j.

Undominated item class $\mathbf{R_j}$

$$\begin{cases} R_1 = \{a_{1,1},\ a_{2,1},\ a_{3,1}\} \\ R_2 = \{a_{1,2},\ a_{2,2}\} \\ R_3 = \{a_{1,3},\ a_{2,3}\} \\ R_4 = \{a_{1,4},\ a_{2,4},\ a_{3,4}\} \end{cases} \quad \begin{array}{c} \textit{Update} \\ \textit{matrix } A \\ \implies \end{array}$$

$$\begin{array}{c} \textbf{Matrix A} \\ \textbf{op}_1 \\ \textbf{op}_2 \\ \textbf{op}_3 \end{array} \begin{array}{c} \begin{array}{cccc} \textbf{sh}_1 & \textbf{sh}_4 & \textbf{sh}_7 & \textbf{sh}_8 \end{array} \\ \left[\begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & \boxed{1} & \boxed{1} & 1 \\ \boxed{1} & 0 & 0 & \boxed{1} \end{array}\right] \end{array}$$

Step 3 : a- Find the profit to weight ratio 'slope' for each shift from the best plan.

E.g., for a shift from $a_{3,1}$ to $a_{2,1}$, the utility loss is $ul_1(a_{3,1}, a_{2,1}) = (u_{3,1} - u_{2,1}) \times size_{sh_{\tau_{sc,o(1)}}} = (u_{3,1} - u_{2,1}) \times size_{sh_1} = (0,7 - 0,65) \times 148 = 7,4$, the frame gain is $fg_1(a_{3,1}, a_{2,1}) = (s_{3,1} - s_{2,1}) = 148 - 68 = 80$, and the utility-to-size ratio $\Upsilon_1(a_{3,1}, a_{2,1}) = \dfrac{ul_1(a_{3,1}, a_{2,1})}{fg_1(a_{3,1}, a_{2,1})} = \dfrac{7,4}{80} = 0,0925$.

$$R_1 = \begin{cases} a_{3,1} \longrightarrow a_{2,1} = \begin{pmatrix} ul_1(a_{3,1}, a_{2,1}) = 7,4 \\ fg_1(a_{3,1}, a_{2,1}) = 80 \end{pmatrix} \Upsilon_1(a_{3,1}, a_{2,1}) = 0,0925 \\ a_{3,1} \longrightarrow a_{1,1} = \begin{pmatrix} ul_1(a_{3,1}, a_{1,1}) = 29,6 \\ fg_1(a_{3,1}, a_{1,1}) = 148 \end{pmatrix} \Upsilon_1(a_{3,1}, a_{1,1}) = 0,1999 \end{cases}$$

$$R_2 = \left\{ a_{2,2} \longrightarrow a_{1,2} = \begin{pmatrix} ul_2(a_{2,2}, a_{1,2}) = 31 \\ fg_2(a_{2,2}, a_{1,2}) = 40 \end{pmatrix} \Upsilon_2(a_{2,2}, a_{1,2}) = 0,7750 \right.$$

$$R_3 = \left\{ a_{2,3} \longrightarrow a_{1,3} = \begin{pmatrix} ul_3(a_{2,3}, a_{1,3}) = 24 \\ fg_3(a_{2,3}, a_{1,3}) = 90 \end{pmatrix} \Upsilon_3(a_{2,3}, a_{1,3}) = 0,2667 \right.$$

$$R_4 = \begin{cases} a_{3,4} \longrightarrow a_{2,4} = \begin{pmatrix} ul_1(a_{3,4}, a_{2,4}) = 23,4 \\ fg_1(a_{3,4}, a_{2,4}) = 160 \end{pmatrix} \Upsilon_4(a_{3,4}, a_{2,4}) = 0,1462 \\ a_{3,4} \longrightarrow a_{1,4} = \begin{pmatrix} ul_4(a_{3,4}, a_{1,4}) = 82,8 \\ fg_4(a_{3,4}, a_{1,4}) = 180 \end{pmatrix} \Upsilon_4(a_{3,4}, a_{1,4}) = 0,4600 \end{cases}$$

b- Order the slopes in an increasing order

$\Upsilon_1(a_{3,1}, a_{2,1}) < \Upsilon_4(a_{3,4}, a_{2,4}) < \Upsilon_1(a_{3,1}, a_{1,1}) < \Upsilon_3(a_{2,3}, a_{1,3}) < \Upsilon_4(a_{3,4}, a_{1,4}) < \Upsilon_2(a_{2,2}, a_{1,2})$

c- Begin iterative shifts until the condition $Ssum \leq maxfnb$ is satisfied.

- Execute the first shift : $a_{3,1} \longrightarrow a_{2,1}$

Mat. U	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1}$	0,50	0,80	0,75	0,34
$\mathbf{op_2}$	0,65	1,00	0,90	0,67
$\mathbf{op_3}$	0,70 ↑	0,40	0,20	0,80

Mat. S	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1}$	0	0	0	0
$\mathbf{op_2}$	68	40	90	20
$\mathbf{op_3}$	148 ↑	155	160	180

Update matrix A $\Longrightarrow$

Matrix A	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1}$	1	1	1	1
$\mathbf{op_2}$	1	1	1	1
$\mathbf{op_3}$	0 ↑	0	0	1

$Ssum = 378 > maxfnb = 288 \Longrightarrow$

Execute the next shift.

- Execute the second shift : $a_{3,4} \longrightarrow a_{2,4}$

Mat. U	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1}$	0,50	0,80	0,75	0,34
$\mathbf{op_2}$	0,65	1,00	0,90	0,67
$\mathbf{op_3}$	0,70	0,40	0,20	0,80 ↑

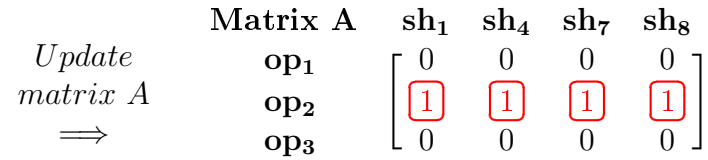
Mat. S	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1}$	0	0	0	0
$\mathbf{op_2}$	68	40	90	20
$\mathbf{op_3}$	148	155	160	180 ↑

Update matrix A $\Longrightarrow$

Matrix A	$\mathbf{sh_1}$	$\mathbf{sh_4}$	$\mathbf{sh_7}$	$\mathbf{sh_8}$
$\mathbf{op_1}$	1	1	1	1
$\mathbf{op_2}$	1	1	1	1
$\mathbf{op_3}$	0	0	0	0 ↑

$Ssum = 218 < maxfnb = 288 \Longrightarrow$

Feasible adaptation plan is found.



Best $\mathbf{ap(sc_1, o_2)} = ((op_2, sh_1),\ (op_2, sh_4),\ (op_2, sh_7),\ (op_2, sh_8))$ such that $\mathbf{GUF(ap(sc_1, o_2))} = \mathbf{515.8005}$

FIGURE E.1 – Example illustrating the heuristic steps.

Due to the small number of shots to be adapted in this example ($|Adp\mathcal{SH}_{sc,o}| = 4$), it was possible to execute an exhaustive search to identify the optimal adaptation plan. First, we found all the adaptation operations combinations, that is $3^4 = 81$, whereas only 42 combinations satisfy the owner constraint. Then, we calculate the global utility value of the 42 adaptation plans and sort the value in an increasing order to identify the optimal adaptation plan. Actually, the heuristic found the optimal adaptation plan. Indeed, this can happen for a low number of shots to be adapted. However, this is very improbable with a high number of $|Adp\mathcal{SH}_{sc,o}|$ in a scene, as we will discuss in Section E.2.

Theoretically, the run-time complexity of the presented heuristic is linearithmic ($O(n \log n)$) where $n = (|\mathcal{OP}'| - 1) \times |Adp\mathcal{SH}_{sc,o}|$. This complexity is due to the sorting of the slopes, which has the highest-order term. Indeed, sorting algorithm cannot perform better than ($O(n \log n)$) in the average or worst case. For our heuristic, the worst case would be to execute $|\mathcal{OP}'| - 1$ shifts for each class R_j, $j = 1, \ldots, |Adp\mathcal{SH}_{sc,o}|$. Section E.2 presents experiments on randomly generated problems, together with simulation results, validating this theoretical evaluation and verifying the efficiency of the presented heuristic.

E.2 Heuristic Evaluation

In this section, we evaluate the performance of the heuristic for finding the adaptation plan that satisfies both end-user and owner constraints, and has a global utility value close to the one of the optimal plan. To begin with, we describe the simulations that we used to analyze the performance of our heuristic in terms of its running time with respect to the number of shots. Then, we proceed to evaluate the quality of the generated plans by comparing them to that of the optimal plans obtained by an exhaustive search algorithm, which remains possible when the number of shots is low.

E.2.1 Experimental Setup

The experiments were run on a HP Pavilion *dv*3 with 2.53 GHz Core *i*5 processor and 6 GB of RAM. We developed the algorithm presented in Section E.1 in Java version 1.6.0_25 and ran it on Windows 7 as operating system. For the input of the algorithm,

we randomly generated ten matrices S, with an increasing number of shots to be adapted $|Adp\mathcal{SH}_{sc,o}|$ from 5 to 50 with step size 5. The elements $s_{i,j}$ of these matrices consists of the number of the frames to be dropped given an adaptation operation (i.e., $Drop-Object$, $Drop-SOFS$, $Drop-Shot$). With respect to their value, $s_{i,j}$ are gradually inherited from the previous matrix to the next one. For instance, a matrix S_4 consists of 20 shots, such that the elements $s_{i,j}$ for the first 15 shots are the same as the ones of matrix S_4. For $i = 1, \ldots, 3, j = 1, \ldots, |Adp\mathcal{SH}_{sc,o}|$, the generation of the $s_{i,j}$ values depend on the adaptation operation. For $Drop-Shot$, $s_{3,j}$ is equal to the size of the shot. We define the size of a shot to be a random number between 50 and 250, inclusively. With these parameters and a frame rate of 25 fps, the length of a scene can vary from 10 secs (2 sec per shot for a scene of 5 shots) up to 8 min 20 secs (10 sec per shot for a scene of 50 shots). This range is a realistic one for narrative videos. We remind the readers that matrix S corresponds to the shots to be adapted within a scene. For $Drop-SOFS$, we define $s_{2,j}$ to be randomly distributed in $[(\frac{s_{3,j}}{2})-25, (\frac{s_{3,j}}{2})+25]$. Thus, if the size of the shot is 50, $s_{2,j}$ takes a value between 0 and 50. Finally, $s_{1,j}$ is always equal to zero since the $Drop-Object$ operation doe not yield removal of frames.
With respect to utility values, twenty matrices U with randomly distributed values in $]0,1[$ are generated for each of the ten matrices S. This random generation of utility values enables us to confront the algorithm to a large spectrum of cases with respect to the properties of the utility matrix. In a similar way to matrices S, the utility values $u_{i,j}$ are gradually inherited from the previous matrix to the next one.
Finally, we define the owner constraint such that the number of dropped frames should be lower than a fixed percentage x% of the size of the scene in frames. With no loss of generality, we consider that all shots of the scene should be adapted. Thus, the condition $maxfnb$ is calculated by : $maxfnb = \frac{x}{100} \times \sum_{j=1}^{|Adp\mathcal{SH}_{sc,o}|} s_{3,j}$.

E.2.2 Analysis of the Execution Time of the Heuristic

The aim of this experiment is twofold : 1) analyze the time required by the heuristic to select the adaptation plan with respect to the number of shots, and 2) compare it with the theoretical complexity of $(O(n \log n))$ established in Section E.1. We fix the percentage of the dropped frames to $x = 20\%$, and the number of items per class to $|\mathcal{OP}'| = 3$. For each couple S and U, we launch 100 iterations and then calculate the average of the time in nanoseconds (see Figure E.2). This process is repeated over every S and the twenty related matrices U. The average of the average values as well as the standard deviation, are computed and represented in the graph.

The run-time complexity of the heuristic is depicted in Figure E.3. The blue lozenge indicates the average of the average values of each class size (e.g., $Avg-Size5$ in Figure E.2). The red error bar refers to the Standard Deviation (STD) of the 20 average values of each class. As depicted in the figure, the computational results demonstrate that, even with a large number of classes (i.e., number of shots to be adapted), the feasible adaptation plan is reached in a fraction of seconds (i.e., less than 100 ms). Furthermore, another conclusion regarding the time complexity with respect to the number of slopes can be drawn. As previously discussed in Section E.1, a slope is the utility-to-size ratio obtained by shifting from an operation to another. We also argued that theoretically the ordering of these slopes is the most expensive step in the heuristic,

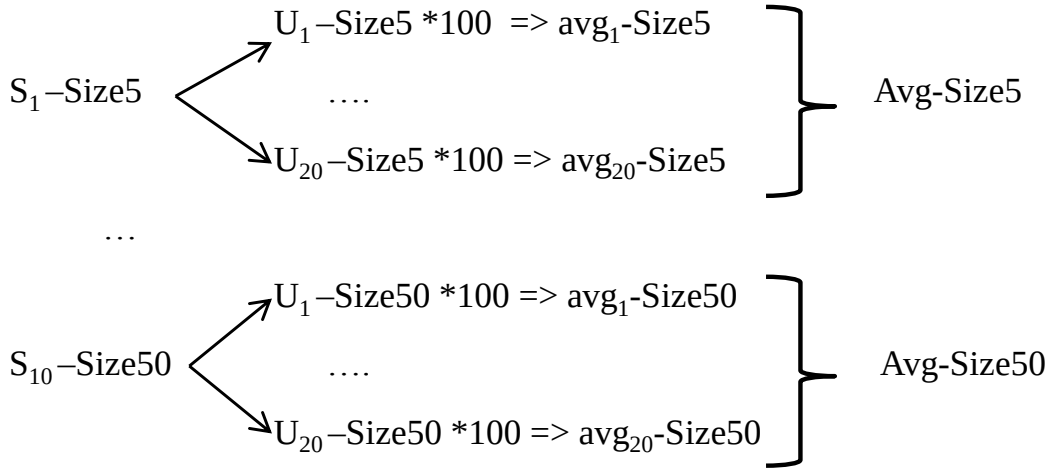


FIGURE E.2 – Data to be represented in the graph.

and it is the reason why the run-time complexity is linearithmic. According to the graph in Figure E.3, we observe that the time increases in a linear manner with respect to the growth of the number of slopes. This result can in fact be explained. Indeed, we use the insertion sort algorithm in our implementation to sorts the slopes. The average case performance of this algorithm is quadratic ($O(n^2)$). Moreover, quadratic algorithms are generally known to be practical for use only on relatively small problems. In our experiment, the number of tested shots is 50 and the number of slopes to be sorted is $n = (|\mathcal{OP}'| - 1) \times |Adp\mathcal{SH}_{sc,o}| = 2 \times 50 = 100$ in the worst case. Furthermore, the sorting algorithm took as an input the list of slopes that was partially sorted. Therefore, since the experimentation is conducted on a small set of data already partially sorted, this explains the linear behavior of the insertion sort in our case.

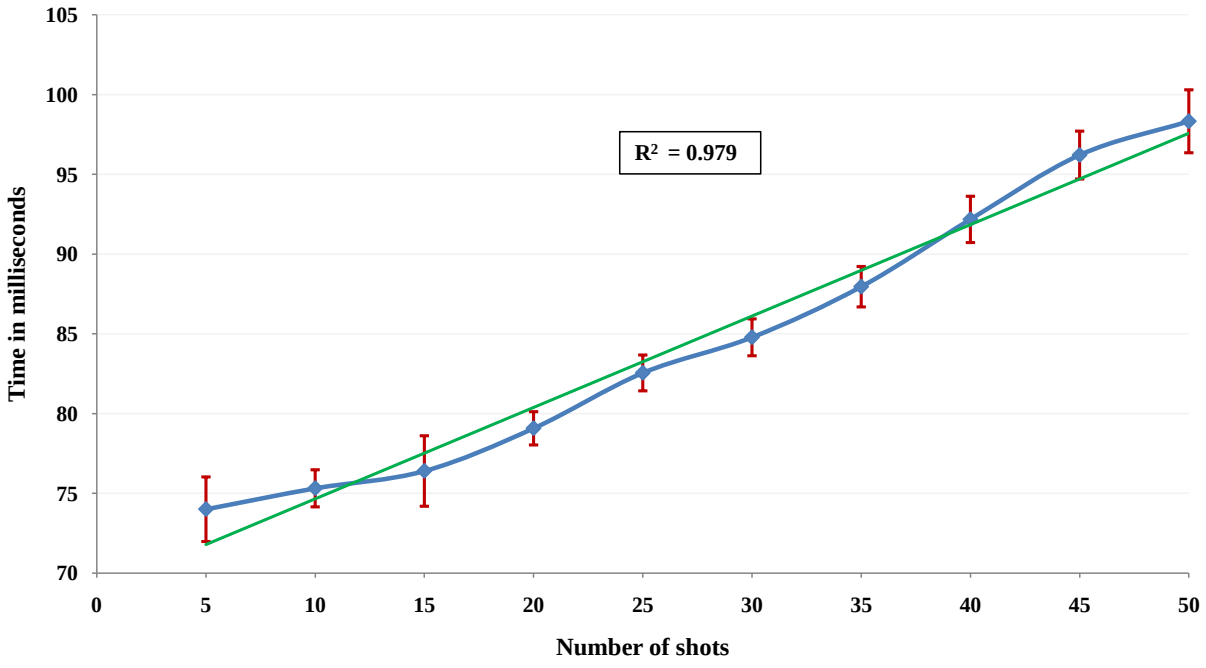


FIGURE E.3 – Run-time complexity of the heuristic.

E.2.3 Quality Evaluation of the Generated Adaptation Plan

In order to evaluate the efficiency of the heuristic in approximating the optimal solution, we implemented the exhaustive search algorithm and applied it for the same matrices with the same owner constraint. We limited the evaluation to matrices that have 5, 10 and 15 shots. This restriction is due to the number of adaptation plans produced by the exhaustive searching that grows exponentially with the increase of the number of shots. Indeed, the execution of exhaustive searching requires a lot of memory, which cannot be allocated by the Java virtual machine for matrices beyond 15 shots.

To begin, for each couple S and U, we launch the exhaustive search algorithm and compute only the global utility value of all possible feasible adaptation plans. Then, we sort these values in a decreasing order and find the optimal adaptation plan, *oap*, which is the plan with the highest global utility value. It is worthwhile to note that we kept all the feasible adaptation plan instead of only keeping the optimal one, in order to compare the adaptation plan generated by our heuristic denoted ap_h against the optimal one. This comparison is done with respect to the following measurements :

- **Accuracy** the distance in number of plans between *oap* and ap_h. It is defined as follows :

$$accuracy = \Delta\,(oap,\ ap_h) \tag{E.3}$$

- **Percentage of the global utility loss** denoted by $pgul(oap,\ ap_h)$ and defined as :

$$pgul(oap,\ ap_h) = 100 \times \left(\frac{GUF_{oap} - GUF_{ap_h}}{GUF_{oap}}\right) \tag{E.4}$$

For instance, Figure E.4 shows the global utility value GUF of *oap* and ap_h for a matrix S of 15 shots along with twenty different matrices U. As can be observed from the figure, the plan produced by our algorithm is very close to the *oap* and often succeeds in generating it.

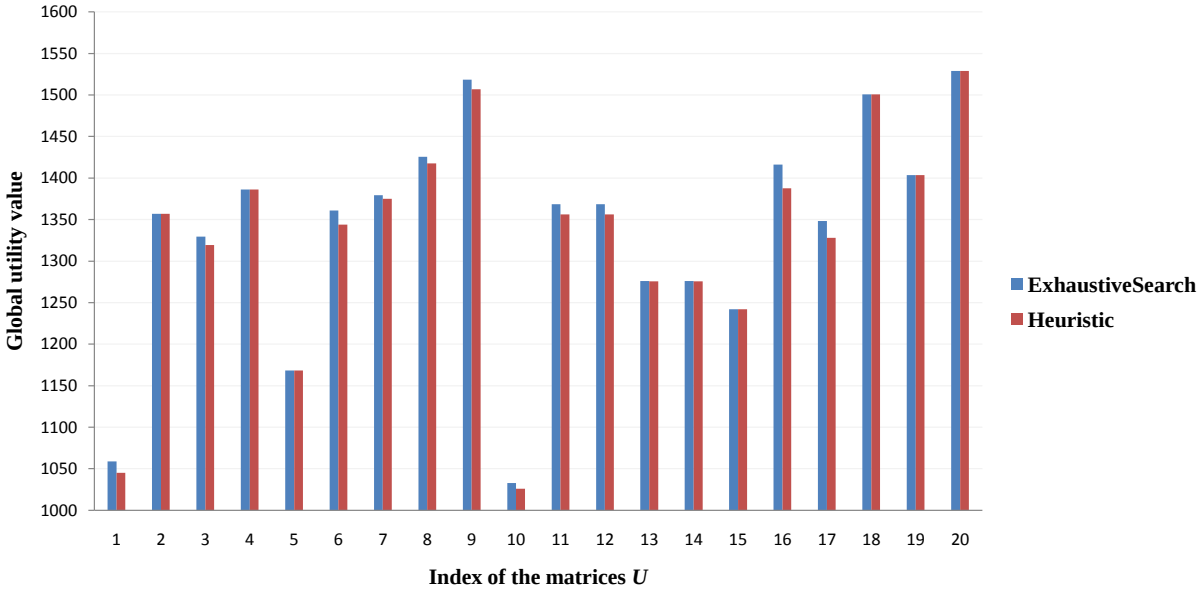


FIGURE E.4 – Accuracy evaluation of the heuristic for a matrix of 15 shots.

This comparison process is repeated over the three matrices S and the twenty related matrices U. The result of this evaluation is depicted in Table E.1. For the percentage of the global utility loss *pgul* measurement, we computed its variation range (i.e., Min. and Max.) as well as its average. For the accuracy measurement, we calculated the percentage of pairs (S, U), which result in the optimal plan by applying the heuristic (i.e., accuracy$= 0$). Similarly, we calculated the percentage of pairs resulting in a feasible plan that is within the five best plans (accuracy≤ 5).

TABLE E.1 – Efficiency results of the heuristic

	pgul			**% of pairs** (S,U) **having**	
	Min.	*Max.*	*Avg.*	$accuracy = 0$	$accuracy \leq 5$
Matrix of 5 shots	0	12.12	1.502	70	95
Matrix of 10 shots	0	6.13	1.457	45	80
Matrix of 15 shots	0	1.28	0.550	35	70

As can be observed from the table, the average of the *pgul* shifts from a higher to a lower value rates (i.e., 1.502 to 0.550) and the percentage of the accuracy decreases (i.e., 70% to 35%) with respect to an increase in the number of shots per matrix (i.e., 5 shots to 15 shots). The reason for this is that the higher the number of shots, the higher the number of adaptation plans generated by an exhaustive search. Thus, the probability of having heuristic adaptation plans with a close global utility value is lower. Moreover, even though the accuracy of the heuristic for choosing the *oap* decreases, the probability of the ap_h being between the first five adaptation plan is still higher than 70%.

Annex F

Evaluation of the Execution time of Adaptation Operations

This Annex describes the experimentations that have been conducted to evaluate the execution time of the three adaptation operators : $Drop - Shot$, $Drop - SOFS$ and $Drop - Object$. All the experiments were run in Java version 1.6.0_25 on a HP Pavilion $dv3$ with 2.53 GHz Core $i5$ processor with 6 GB of RAM, and on Windows 7 as an operating system.

F.1 Execution Time of $Drop - Shot$

For the implementation of the $Drop - Shot$ adaptation operator, we used the scripting engine of Avidemux [17]. Here, we implemented an algorithm that takes a video and the interval of the shot that needs to be removed as an input, creates the script for the Avidemux' engine, and returns the adapted video as an output.

	Shot Interval	Shot Length	Time Average (sec)	Standard Deviation (sec)
Shot 1	0-122	123	0.348	0.017
Shot 2	123-180	58	0.346	0.013
Shot 3	181-265	85	0.347	0.017
Shot 4	266-559	294	0.344	0.012
Shot 5	560-700	141	0.346	0.016
Shot 6	701-764	64	0.345	0.015
Shot 7	765-891	127	0.344	0.012
Shot 8	892-940	49	0.346	0.015
Shot 9	941-1091	151	0.346	0.015

TABLE F.1 – Time execution of $Drop - Shot$ algorithm with respect to the shot length in frames.

Given this $Drop - Shot$ algorithm, we conducted experiments to evaluate the time required for removing a shot with respect to its length in frames. To this end, we prepare a video of 9 shots, each of different length each (see Table F.1). The video has a resolution of 320×240 and a frame rate of 25 fps. For each shot, we launch 100 iterations of the $Drop - Shot$ algorithm over the original video, and then calculate the average of the execution time and the associated standard deviation in seconds.

17. http://www.avidemux.org/

According to the experimental results in the Table F.1, we observe that the execution time of the $Drop - Shot$ algorithm is independent of the length of a shot, and it is a constant.

F.2 Execution Time of $Drop - SOFS$

For the implementation of the $Drop - SOFS$ adaptation operator, we modify the previous algorithm to $Drop - SOFS$ algorithm so that it can take a list of intervals of frame sequences to be removed, as an input. Since $Drop - SOFS$ is always executed in a shot, then the frame sequence intervals should belong to one shot of a video. The same video described in Section F.1 was used in this experiment. More precisely, we execute the $Drop - SOFS$ algorithm on $Shot5$, which consists of 4 frame sequences (see Table F.2).

	Shot 5			
Frame Sequence	**FS_1**	**FS_2**	**FS_3**	**FS_4**
Frame Sequence Interval	560-584	585-614	615-635	636-700
Frame Sequence Length	25	30	21	65

TABLE F.2 – Description of the setup data for $Drop - SOFS$ algorithm.

Accordingly, we prepare 9 lists of frame sequences to be removed (see Table F.3). For each list, we launch 100 iterations of the $Drop - SOFS$ algorithm over the original video, and then calculate the average of the execution time and the associated standard deviation in seconds. According to the experimental results in the Table F.3, we observe that the execution time of the $Drop - SOFS$ algorithm is independent of the number and length of frame sequences, and it is a constant.

	List of Removed Frame Sequences	Time Average (sec)	Standard Deviation (sec)
1	**FS_1**	0.344	0.016
2	**FS_2**	0.344	0.015
3	**FS_3**	0.344	0.012
4	**FS_4**	0.344	0.010
5	**FS_1, FS_3**	0.344	0.020
6	**FS_1, FS_4**	0.349	0.014
7	**FS_2, FS_4**	0.345	0.016
8	**FS_1, FS_2, FS_3**	0.343	0.013
9	**FS_1, FS_2, FS_4**	0.345	0.015

TABLE F.3 – Time execution of $Drop - SOFS$ algorithm with respect to the number and length of frame sequences.

F.3 Execution Time of $Drop - Object$

For the implementation of the $Drop - Object$ adaptation operator, we use the inpainting algorithm described in [95]. The authors of this paper present an interactive system based on an intuitive user-friendly interface for removing undesirable objects in images. To remove an object in an image, a user selects the object by simply pinpointing it with the mouse cursor. Afterwards, a hole-filling technique is employed using the neighbourhood pixels from the background. More details about the algorithm are given in [95]. The reason for choosing this algorithm lies in its good performance for complex objects on the one hand, and the availability of the code on the other hand.

Given this $Drop - Object$ algorithm, we conducted experiments to evaluate the time required for the inpainting process with respect to the size of the image and the spatial regions of the object to be inpainted. The data set comprises 19 images having a resolution of 1000×750. The color and texture characteristics of object and background were chosen to be homogeneous as depicted in figure F.1. The object is represented by a black rectangle in the image. We prepare the first image such that the object size is 0.25% of the size of the image (i.e., 1900 pixels). For the rest of the images, we progressively increase the size of the object by increments of 0.25% of its original size.

FIGURE F.1 – Color and texture characteristics of object vs. background.

For every image, we launch 200 iterations of the inpainting process, and then calculate the average of the execution time and the associated standard deviation in seconds. Figure F.2 illustrates the result of the execution time of the inpainting process, which is described in Table F.4. The green triangle indicates the value resulting from the fitted model, the blue square indicates the execution time value, and the green line to the polynomial regression. The standard deviation values are negligible compared to the execution time, and vary between 0.002 and 0.1 second. Therefore, their representation on the graph was not possible given the axis scale. According to the graph in the figure, we observe that the execution time of the inpainting algorithm increases in a polynomial manner $O(n^2)$ with respect to the size of objects. Indeed, the result of the polynomial regression conforms with the polynomial function of degree two with a correlation coefficient equal to 0.98.

% of the Object Size in the Image	Time Average (sec)	Standard Deviation (sec)
0.25	0.640	0.002
0.40	0.783	0.002
0.57	0.986	0.002
0.78	2.393	0.085
1.01	3.200	0.019
1.28	3.753	0.015
1.58	4.254	0.034
1.92	7.106	0.052
2.28	6.788	0.024
2.68	7.999	0.013
3.10	9.593	0.023
3.56	12.830	0.058
4.05	12.176	0.100
4.58	15.481	0.048
5.13	21.192	0.080
5.72	28.428	0.086
6.33	30.846	0.063
6.98	33.133	0.076
7.66	35.780	0.023

TABLE F.4 – Time execution of $Drop - Object$ algorithm with respect to the size of objects in pixels.

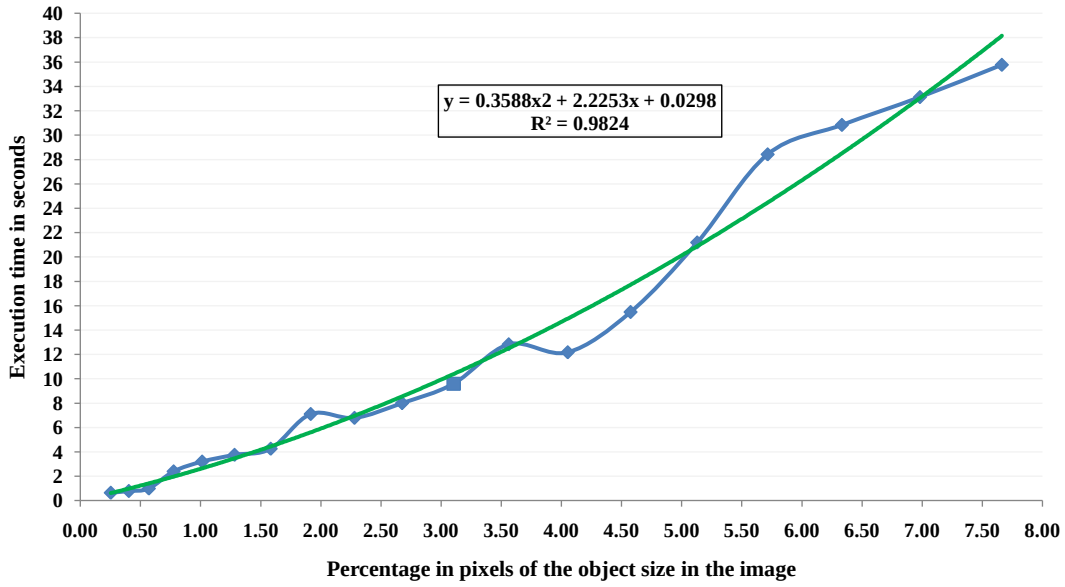


FIGURE F.2 – Curve of the time execution of $Drop - Object$ algorithm.