HABILITATION À DIRIGER DES RECHERCHES
DE
L'UNIVERSITÉ PIERRE ET MARIE CURIE

HABILITATION À DIRIGER DES RECHERCHES

Spécialité : **Sciences pour l'ingénieur**

présentée par

Cyril CONCOLATO

Interactive Media Streaming

Soutenue le **25/10/2016** devant le jury composé de

M.	Vincent	**CHARVILLAT**	Professeur, Université de Toulouse	Rapporteur
M.	Dick	**BULTERMAN**	Professeur, Vrije Universiteit Amsterdam	Rapporteur
M.	Carsten	**GRIWODZ**	Professeur, Universitetet i Oslo	Rapporteur
M.	Pierre	**SENS**	Professeur, Université Paris 6	Examinateur
M.	Gwendal	**SIMON**	Maître de Conférences, HDR, Telecom Bretagne	Examinateur
M.	Christian	**TIMMERER**	Associate Professor, Universität Klagenfurt	Examinateur
M.	Jean-Claude	**DUFOURD**	Directeur d'études, Telecom ParisTech	Examinateur

Contents

Chapter 1

Foreword

At the time of writing of this report, I have been carrying research activities for the past 15 years, mostly at Telecom ParisTech, formerly Ecole Nationale Supérieure de Télécommunications, in Paris, France.

Telecom ParisTech is a public institution and as such dissemination is a very important part of the research. Dissemination often takes the form of published papers. I have published so far 60 papers in journals or conference proceedings. I have also applied for more than 18 patents. A list of my published papers available online in my publications page[1] and many are listed in the bibliography chapter of this report. Figure 1.1 presents a graphical overview of some of my publications, since my PhD, between 2008 and 2015, along a timeline, highlighting the topics I have covered and the links between publications. These topics will be covered in this report.

The impact of the dissemination of a researcher is often measured with paper citations. According to my Google Scholar's profile[2], I've been cited 423 times. This does not unfortunately measure the impact of my contributions to open-source software, which is a particularity of my research activity and that I consider a complementary and important dissemination activity. Most, if not all, of my research leads to publication of open-source software, in particular within GPAC[3], a multimedia platform developed at Telecom ParisTech. My GitHub profile page[4] indicates a total of 568 contributions to open-source software in the last year. My OpenHub profile page[5] indicates

---

[1] Full list of publications `https://concolato.wp.mines-telecom.fr/publications/`

[2] Google Profile page `https://scholar.google.fr/citations?user=ySXws0oAAAAJ&hl=fr&oi=ao`

[3] Website of the GPAC project `http://gpac.io`

[4] Personal GitHub page `https://github.com/cconcolato`

[5] Personal OpenHub page see https://www.openhub.net/p/gpac/contributors/2598440182341747

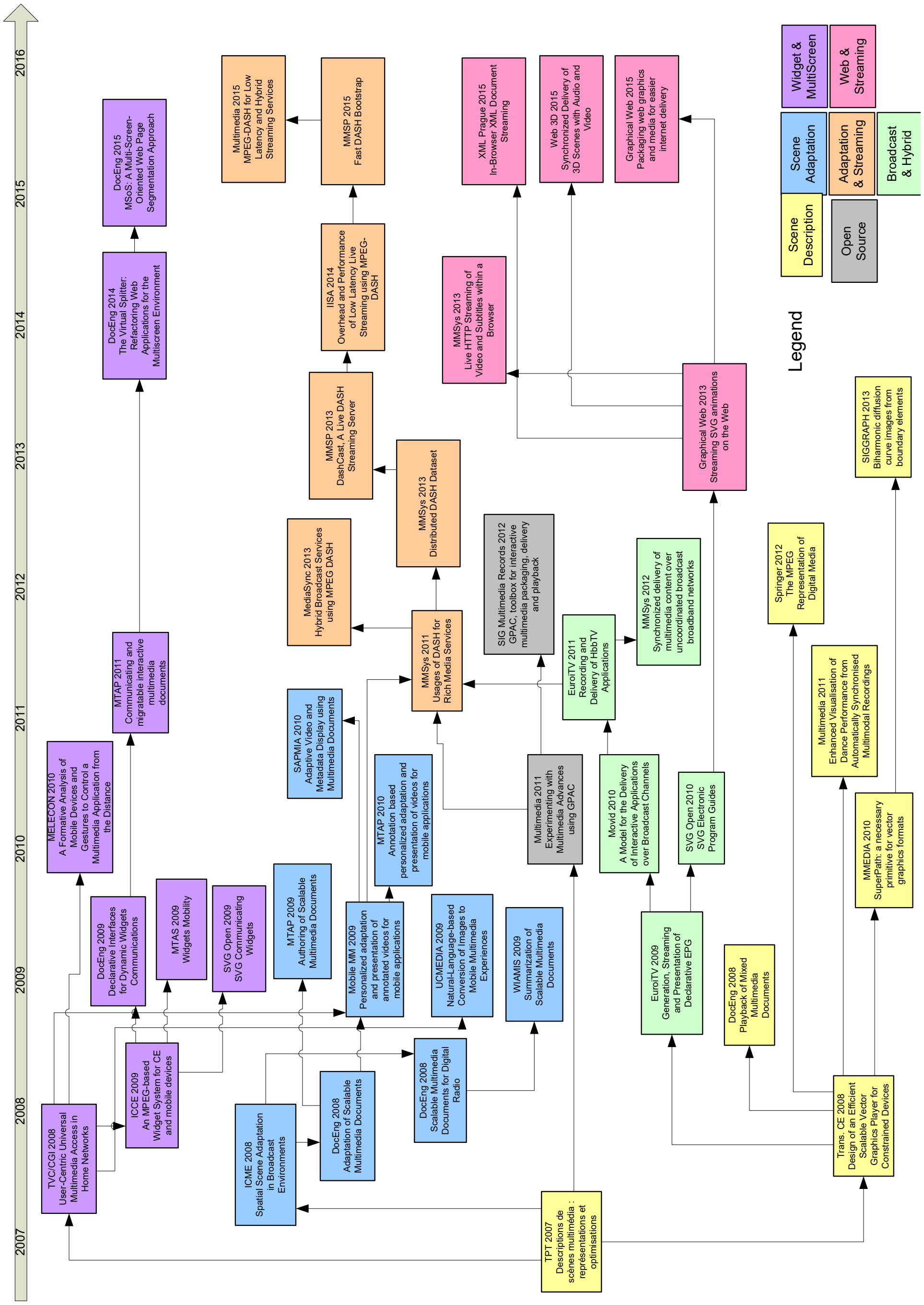


Figure 1.1: Timeline of my academic publications

a total of about 700 contributions to GPAC since the beginning of these statistics. A confirmation that my open source activity is important for my research is that my most cited paper is the first paper presenting GPAC at the ACM Multimedia Conference [44].

When analyzing someone's research record, it is also interesting to see who he/she publishes with. I've published my academic papers with 81 different persons from 20 different institutions, but mostly research centers and few universities. This highlights an important aspect of my research activities, that most of them have been carried through collaborations with companies, rather than with other universities. This is the consequence I believe of two characteristics. First, most of my research has been funded either by the French research agency, by the European union, or directly by private companies, through collaborative projects, which put me in contact more with companies, in particular small and medium enterprises, than with universities. I find this contact with companies very important for my research (and my teaching activities), keeping it in sync with the industry. Second, an important component of my research activity is the participation to standardization activities, mostly within the Motion Picture Experts Group (MPEG) and the World-Wide Web Consortium (W3C). It is hard to analytically measure the impact of such activity, but I've been the contributor and editor of many standards including some important ones: the Scalable Vector Graphics standard (SVG), the ISO Base Media File Format (ISOBMFF), and to a lesser extent the Hyper-Text Markup Language (HTML). A complete list of my contributions to MPEG is also reported on my publications page[1], and Figure 1.2 proposes a compact visualization of the different contributions over time per technology type, again only since my PhD. Other contributions to other standardization bodies like W3C are hard to trace, being made by emails, phone conferences or physical meetings and not represented, but none the less important.

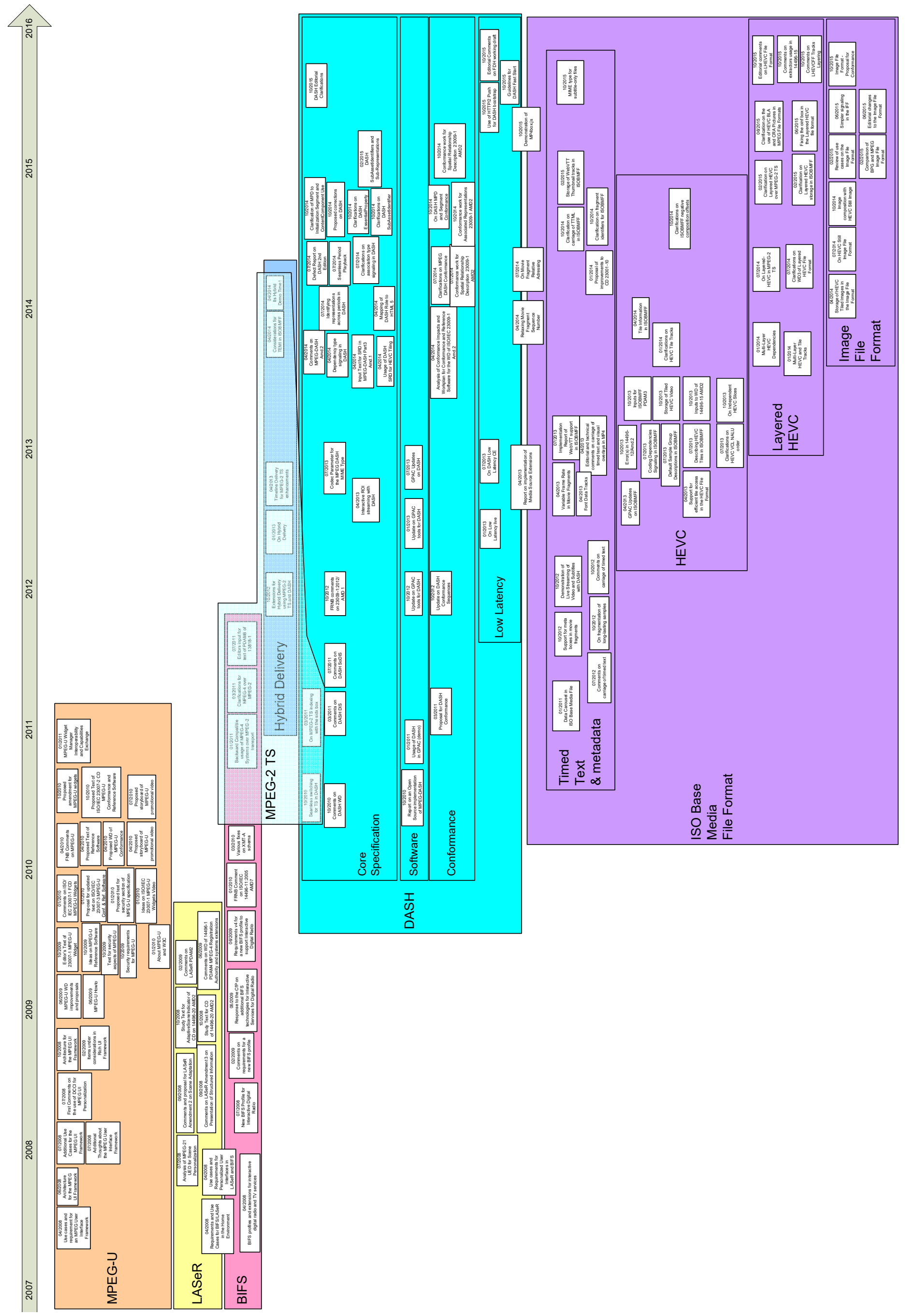


Figure 1.2: Timeline of my MPEG publications

Chapter 2

Introduction

This report presents my latest research work, i.e. since I defended my PhD in July 2007. My PhD thesis was about Multimedia Scene Representations. It was about how to represent, compress, deliver and render the description of the spatial, temporal and interactive organization of the media elements in a multimedia presentation. During that period, I became interested in many other aspects related to multimedia. These aspects are sometimes called *Systems* aspects, in particular in the MPEG terminology. They relate to all the technologies, outside of compression technologies, required to offer multimedia services. They range from basic media synchronization, to media packaging, to delivery methods such as streaming, to content protection, and of course to media presentation including interactivity. I have carried research activities in all these aspects. Most, but not all, are covered in this report.

At the end of my PhD work, I started following and participating to W3C standardization activities, in particular to the SVG standard. This was my real first step into Web technologies. SVG at that time was seen as a promising technology for web applications, mixing graphics and text in an interactive and animated manner, in particular based on Synchronized Multimedia Integration Language (SMIL). Since then, the Web world has evolved. HTML and Cascading Style Sheets (CSS) have adopted many of the features of SVG to the extent that now HTML, specifically HTML5, is the undisputed standard for developing web multimedia applications, for all platforms, from mobile, to tablet, to PC, to TV. Chapter 3 presents the evolution of my research work related to Web Applications. It can be viewed as a continuation of my earlier work on Scene Description, but transitioning from old technologies such as MPEG-4 BIFS, to SVG, and finally to HTML directly.

During my PhD, I also started studying the problems of adaptation. I was interested in vector graphics adaptation and also more generally in media adaptation, in the context of different collaborative projects. Since then I pursued this research activity but broadening its scope, studying the concepts of scene adaptation in broadcast environment, of adaptation and streaming, and of web adaptation. Chapter 4 of this document presents these works.

As indicated earlier, the long-awaited convergence of technologies between the Web world, the mobile world and even the television world has begun from an application perspective, with HTML5. But it is also happening - to a lesser extent probably - in the media world with codecs such as the Advanced Video Coding (H.264—AVC) and Advanced Audio Coding (AAC) standards or packaging or delivery formats such as ISOBMFF or Dynamic Adaptive Streaming over HTTP (DASH). These codecs or formats are being supported by almost all browsers on all platforms, and at the protocol level where the IP protocol is almost ubiquitous. This opens up new perspectives in the way media can be consumed. Chapter 5 and 6 of this report is about my research work on delivery of media and web applications in this convergent world.

Finally, Chapter 7 presents a conclusion of this report and some perspectives I envisage to investigate in the future.

Chapter 3

From Scene Descriptions to Web Applications

The term *multimedia* is often used to describe the simple use of audio and video content in the same presentation. Sometimes even the simple streaming of video content is also referred to as multimedia. In my research, *multimedia* actually extends to the use of multiple types of media elements, such as audio, video, text, 2D or 3D graphics, fonts in a complex presentation. Such mixed content requires a description of how the media elements are used in the presentation. Such description can be in the form of textual, declarative mark-up languages such as HTML, SMIL or SVG; of binary, compressed, declarative languages such as MPEG-4 Binary Format for Scenes (BIFS), MPEG-4 Lightweight Applicatino Scene Representation (LASeR), Adobe Flash; or even entirely described using programmatic languages such as JavaScript. This type of description is what I called in my PhD work: the Scene Description. During the period covering my PhD, I proposed to structure Scene Descriptions along 3 axis: spatially, indicating how the media component are displayed in a 2D/3D environment; temporally, indicating how this organization can evolve over time in particular when driven by a server; and interactively, indicating how this organization can evolve following user interaction. This work naturally continued after the PhD and is presented in this chapter in Section 3.1.

A key aspect associated to multimedia is interactivity. Interactivity, in my research, refers to the fact that the end-user can influence the playback of the multimedia presentation. In practice, however, multimedia presentations actually deployed and consumed by end-users are based on technologies that are not always cleanly structured from an interactive point-of-view and differ from properly structured Scene Descriptions. This is in particular true for multimedia applications based on Web standards. Based on this view of the current practices, part of my research on interactivity evolved into studying Web applications to see how the concepts of Scene Descriptions could be applied to real applications. The second part of this chapter Section 3.2 presents these works. Finally, with the growth of mobile multimedia devices, usages evolved and people started interacting with multiple devices, with multiple screens. My research in this multiscreen environment is reported in the last Section of this Chapter.

3.1 Scene Description Activities

3.1.1 MPEG-4 BIFS

MPEG-4 BIFS was the first Scene Description technology on which I worked. Its importance, in general and in my research in particular, decreased over time to become a low priority topic of research, even some recent works still use it [52], [53]. The end of my research work on this format is materialized by my participation to a book chapter [16] describing the concepts of Scene Descriptions in the MPEG world. After that publication, my work on scene description activities focused mainly on the SVG format and on issues related to the integration of scene descriptions into applications, which I present hereafter.

3.1.2 Mixing Scene Description formats

During my PhD, I worked on algorithms for efficiently rendering SVG content, especially on constrained mobile devices [43]. This work was integrated in the GPAC open source software, developed at Telecom ParisTech. At the same time, the GPAC player was also capable of processing other formats such as MPEG-4 BIFS or 3D languages such as Virtual Reality Modeling Language (VRML) or X3D. After the PhD, I first investigated how these two aspects could be combined. This work lead to a paper in the ACM Document Engineering conference [38] in which we presented that processing multiple scene description languages (i.e. SVG, VRML, Flash), a priori incompatible, was possible and in a consistent manner. We looked in particular into the mixing of graphical primitives, as illustrated in 3.1, and into the combination of two different interactivity models (Web-based and VRML-based). It confirmed that even if there were several formats to describe multimedia presentations, they could be combined in a single model as there were no fundamental incompatibilities between them. This allowed me to teach classes related to scene description formats in a more structured manner, and gives me a way to analyze more recent technologies.

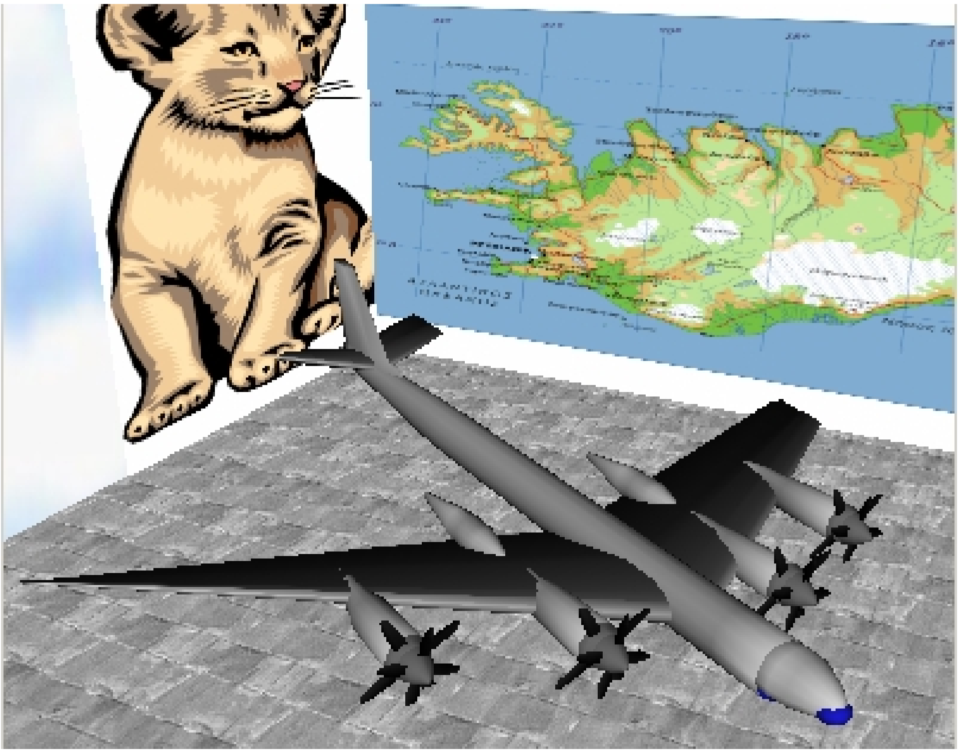


Figure 3.1: Rendering in the GPAC player of mixed 2D/3D content using two different scene description languages: SVG and X3D

3.1.3 Extending the SVG format

The SVG format was the keystone scene description format of my PhD work. Naturally, my interest for this technology continued after my PhD. In particular, I participated to its standardization activities, to several research-oriented and developer-oriented conferences in this area and even organized the SVG Open conference in Paris in 2010[1]. This interest led to the publication of several research papers as presented here.

SVG is mainly a vector graphics format. However, in my experience with the format and in previous attempts to use it as an alternative to Adobe

---

[1] Web site of the SVG Open Conference, Paris 2010, `http://www.svgopen.org/2010/`

Flash [48], I realized that SVG suffered from several representation issues. SVG, like MPEG-4 BIFS, was designed around the concept of *shape*: closed or open, and the *stacking* metaphor. At the opposite, Flash graphics were designed around the concept of planar maps [50]. In a structured world like the Web, a shape has lots of benefits. You can interact with, style or animate any shape. But this has consequences. To describe objects sharing a border, the border is described twice. This leads to larger files, to rendering artifacts (e.g. when applying anti-aliasing methods, as highlighted in [54]) and to potential semantic issues, in particular in map applications. In the conferences on Advances in Multimedia [26], we proposed an extension to the SVG format, called *Super Path*, which allows for reusing paths between shapes. Paths can be used following the orientation with which they were defined or in the reverse orientation. This representation has the advantage of reducing the SVG file size but most importantly of preserving the semantics of the graphics. As a result of this work, the proposed feature has been accepted as part of a *Path* module to the SVG2 standard[2].

My work on SVG also allowed me to collaborate with many researchers in the world. In particular, my participation in the SVG Working Group allowed me to work, between October 2011 and April 2012, at CiSRA (Canon Information Systems Research Australia) as Canon representative in the SVG WG. During that time, I studied the integration of a new graphical primitive into the SVG language, called *Diffusion Curves.* Diffusion Curves, introduced by [51], allow describing rich color variations in a vector graphics image. As opposed to simple primitives describing a uniform color or linear or radial gradients, a diffusion curve describes the color variations along a curve and away from the curve in a powerful, mathematical manner. This allows representing for instance stylized or photo-realistic images in vector graphics form. An example is given in Figure 3.2.

My work with CiSRA has consisted in improving the initial representation of diffusion curves to allow for not only sharp but also smooth color transitions across curves. I also studied several methods for rendering those diffusion curves with a particular focus on animation properties, on stability and rendering performances. The result of this work has been published at SIGGRAPH Asia [11]. Its integration in the SVG language is not finalized yet. Only a simple version called *Coons Patches* is integrated the latest version of the specification[3], but work is still on-going in this area.

---

[2] The SVG 2 Paths module `https://svgwg.org/specs/paths/`

[3] SVG 2 Mesh Gradients specification `https://svgwg.org/svg2-draft/pservers.html#MeshGradients`

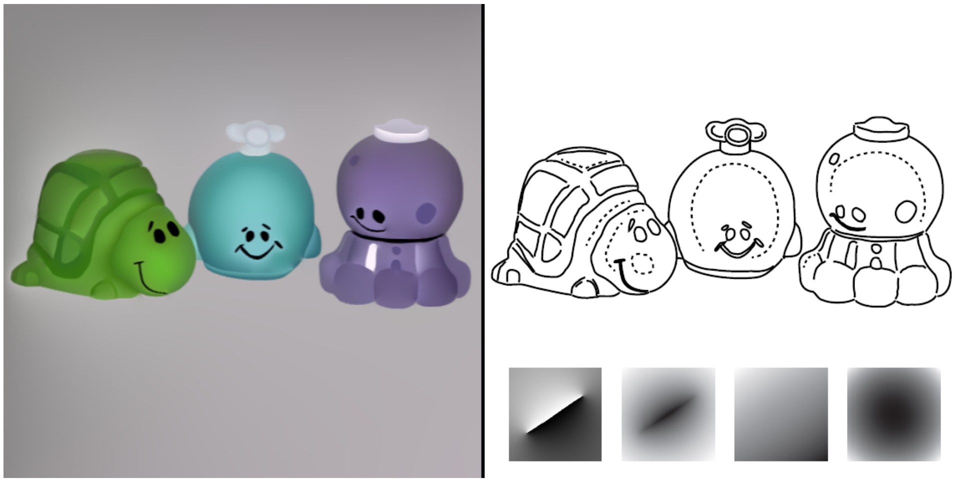


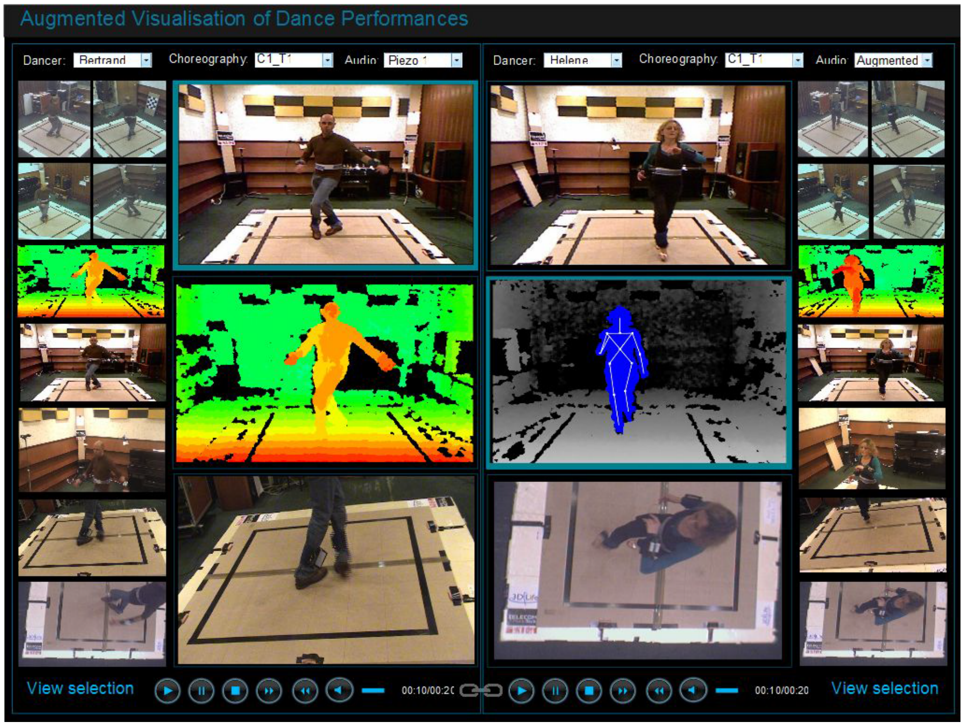
Figure 3.2: Rendering (left) from a set of diffusion curves (right) with color profiles (bottom-right)

3.2 Interactive Media Applications

My work on Scene Descriptions during my PhD was rather focused on the capabilities of the formats and languages to describe scenes. As a follow up after the PhD, I applied the concepts introduced in those formats and languages to build interactive multimedia applications. Some of these applications led to publications in different venues as reported here.

3.2.1 Synchronized and Interactive Streams

As part of a large collaboration within the 3DLife European Project, I worked on the development of an application to evaluate dancer performances [18]. This work was a response to the ACM Multimedia Conference Grand Challenge in 2011. In this work, multiple media streams (texture videos, depth videos, sounds) captured from different sensors (cameras, kinects, microphones) were available as recordings of Salsa dance performances. The idea of the paper was to propose a multimedia application enabling the synchronized visualization of the different performances, in order to perform a subjective evaluation of those performances. The challenges in this application were the inter-stream synchronization and the enrichment of the application by the synchronized display of different statistics together with the video, in an augmented-reality manner. Inter-stream synchronization was done by some co-authors using cross-correlation between the different signals, for instance by analyzing the audio signal envelopes. I have however carried similar work on synchronization in a different context as reported in Section 6.6. My participation in this work was to study and design an application showing the different streams in a synchronized and interactive manner. Tight synchronized rendering was required for displaying the performance of a single dancer together with its statistics. This rendering could be coupled and synchronized when necessary with the same rendering but for a performance of another dancer, for comparison. In this application, I used the SVG language, and in particular, the video and audio tags, which then became part of the HTML5 standard. The player used to render this synchronized content was the player of the GPAC project. A snapshot of the application is given in Figure 3.3.



Dancer: Bertrand
Choreography: C1_T1
Audio: Piezo 1
Dancer: Helene
Choreography: C1_T1
Audio: Augmented

Figure 3.3: Illustration of the enrichment of media elements using scene description

3.2.2 Personalized Interactive Application

Personalization consists in proposing content to the user that is adapted to his/her preferences. Proposing personalized video has been a topic of research of several researchers [83][84]. Together with the University of Ghent and the University of Aachen, as part of the NoE INTERMEDIA, I participated in the development of an end-to-end system for personalized video adaptation and presentation, on mobile applications, guided by automatically generated annotations. This work has been published in Springer's Multimedia Tools and Application journal [25] as well as in the European Symposium on Mobile Media Delivery [30].

The whole architecture of the system was centered on the concept of Region-of-Interest (RoI). A video content is analyzed to detect some relevant objects (e.g. characters, faces, foreground objects). The video is then encoded such that given a target bitrate, the RoI is given more bandwidth than the other parts of the video. The encoded video content is finally packaged together with presentation information that enables interaction with the RoI. This concept of differentiated coding for a region of interest had been adressed already many times, such as [55]. The novelty in that approache consisted in proposing a complete experience with an interactive application.

My contribution in this work was in the packaging and design of the presentation layer. In later publications, I also worked on the encoding aspects of RoI as we will see in Section 5.4. I designed the presentation layer with two important parts: a static part, representing the logic of the application, i.e. the ability to select a RoI and interact with it; and a dynamic part, which contained the time-varying part, i.e. the association of the positions with a given RoI. When working on this application, there were only few choices of technologies to achieve it. I decided to use standard formats such as MPEG-4 BIFS and LASeR. I think the work done for this application is still relevant today, for two reasons. More and more applications require streaming of subtitling or metadata along with audio and video streams. The characteristics of MPEG-4 BIFS and LASeR streams are very close to those of subtitle or metadata streams, in terms of size or a-periodicity. The carriage of such types of streams is very relevant in the context of adaptive streaming, and poses interesting problems on which I worked later, as we will see in Section 5.2. The second aspect that is still relevant today is the use of region of interest for visualizing large images. The Web community has been addressing this use case using the concept of Art Direction[4].

---

[4] Art Direction in Web Pages `https://www.smashingmagazine.com/2016/02/`

The MPEG community has also been working on this aspect with the tiling capabilities of HEVC [56], which will be discussed further in Section 5.4.

`automatically-art-directed-responsive-images-go/`

3.3 Communicating Web Applications

As indicated in the previous section, a significant part of my research is dedicated to interactive and multimedia applications. With the convergence of media, networking and interactivity technologies, it is now possible to design interactive and multimedia applications that run on multiple devices: from mobile phones to televisions. Such applications are called *multi-screen* or Second Screen applications when one screen is a TV [87]. During the period of time covered by this report, I carried research studying the challenges that are posed by the design and use of interactive and multimedia applications running and collaborating on multiple devices. This chapter presents this work. It can be seen as an extension of the previous reported work on Scene Description, but with a more important focus on the relationship between interactivity and networking. Interestingly, this work spans the entire research time from the end of my PhD to today and at the same time the two main types of standard technologies I have worked with are represented: MPEG and Web technologies.

3.3.1 Widgets

My first paper on this topic was presented at the ACM Document Engineering Symposium [31]. In this paper, we defined and reviewed the notion of *widget*, i.e. a small and focused multimedia application that can be run on desktop computers, mobile devices or even TV sets. With my background on Scene Description, widgets could easily be represented using structured multimedia documents. However, we identified that a challenge for those widgets was related to the fact that they required the ability to communicate with remote data sources to change their content. A first issue was that these sources had to be known at authoring time, which was not really convenient. A second issue was that the communication process between the widget and its data sources relied heavily on scripting, making it hard to author, with potential security issues. This was particularly problematic for widgets designed to work in the home network, where the source of information (e.g. a TV set or a phone) was not known at authoring time and possibly different in different home networks. To overcome these problems, in our paper, we designed a declarative mechanism enabling the communication between widgets and their dynamic environment (other widgets, remote data sources). Interestingly, we designed the mechanism to be compatible with existing widgets technologies, usable with script-based widgets (typically web applications) as well as with fully declarative widgets (based on MPEG-4 BIFS). This mechanism was based on the definition of a widget service abstraction and on the description of the interface between the widget and the external services using that abstraction. In our approach, this description is resolved at run-time by the player depending on the available data sources and other widgets. The proposed description uses declarative tools (based on XML) and is stored in a structured document, called a *manifest*, inspired by the W3C Packaging standard[5]. By declaring in the widget manifest all the interfaces a widget may use, we made it easy for the widget manager to validate the connections before execution and to implement security policies.

We demonstrated this work via a prototype implementation in the open source player from the GPAC project. We showed that this approach could indeed be used with existing multimedia description languages such as SVG or MPEG-4 BIFS. We presented the results of this work during the SVG Open Conference [29], showing how widgets written in the SVG language could benefit from our proposed approach in the home network. We demonstrated SVG widgets running on different devices (TV, phones) and communicating with each other. This work was also presented at the Conference on Mobile Technology, Applications and Systems [32] focusing on how the binding between the widget and network stacks was made, for instance using the UPnP protocol, to achieve widget discovery and mobility.

Finally, a journal paper was published in Springer's Multimedia Tools and Applications journal [20], summarizing all these aspects.

This work was also submitted to MPEG in the course of the new standardization activity related to Rich-Media User Interfaces. It was selected as a basis for the MPEG-U standard, for which I became co-editor. This led to the submission of more than 25 contributions, most of them as part of a collaboration with Samsung R&D. Together with Samsung R&D, we also presented the results of this standardization work during the IEEE Conference on Consumer Electronics [36].

As a side result, using this work, as part of the INTERMEDIA NoE, together with the Universities of Lancaster, of Aachen and the Fraunhofer Institute, we studied and compared different methods to interact with a multimedia application either using mobile devices with hardware buttons, software components (in particular widgets), and gestures. The result of this work was published at the IEEE Mediterranean Electrotechnical Conference [27].

---

[5] W3C Packaging Standard `https://www.w3.org/TR/web-packaging/`

3.3.2 Multi-screen Web Applications

During our previous widget-related work, we soon realized that an interesting challenge related to use of widgets was the design of coherent applications, using widgets, that could be easily distributed across devices. I started doing research in this activity, based on the premise that designing a web application for the desktop was well known, even if this application was composed of widgets. But I realized that designing an application whose widgets were easily extractable and processable on different collaborating devices was a challenge. I started working on this topic during the supervision of a Master student (V. Murga). We then worked on this topic as part of a joint research project called COLTRAM with the Fraunhofer Fokus Institute from Berlin, Germany. The work was mainly carried during the PhD of Mira Sarkis that I supervised with Jean-Claude Dufourd.

The work started with an investigation of how web applications were built and of how they could be split and distributed to different devices. Realizing that most of the applications were designed using JavaScript, we attempted to determine the best split by performing a static analysis of the JavaScript. This turned out to be very difficult and not practical enough. We therefore opted for a run-time model for multi-screen applications based on master/slave relationship. The master application runs the entire JavaScript. A communication channel is established between the master and the slave applications. Only the necessary application changes are forwarded to the slave application and only relevant events are forwared from the slave to the master, to trigger JavaScript calls. With such model, the only split that was needed for distributing the application was at HTML and CSS levels. This work was presented at the ACM Symposium on Document Engineering [5]. Later on, we realized that the HTML split was a task that could be improved. We therefore worked on a multi-screen driven approach of web page segmentation that we also presented in the same venue the next year [9], proposing an automated framework for repurposing web applications for the multi-screen world. Work is still ongoing to improve the layout of the split applications by taking into consideration more carefully its CSS parts, and a paper has been submitted to Springer's Multimedia Tools and Applications journal.

3.4 Conclusion

The field of interactive multimedia applications is a field of rapid evolutions. Technologies change at a fast pace. The SVG format, which was, at some point, seen as the promising technology for interactive multimedia application, has now been swallowed into the Web ecosystem. HTML5, with its ability to mix audio, video, text and graphics content, has become the de-facto technology for writing interactive multimedia applications. This brings benefits (e.g. the ability to reach a large audience with well deployed browsers) and problems (due to the historical workarounds and the lack of structure in the HTML language). I believe that my work on Scene Description formats and my experience in designing interactive multimedia applications are relevant background to solve many research problems of this new HTML5 era. In particular, HTML5 is currently facing challenges such as the ability to easily structure applications by reusing components. Research work is going on the development of the *Web Components* standards[6]. Research is also being carried to improve the responsiveness of HTML application, with Facebook's Instant Articles[7] or Google's AMP (Accelerated Mobile Pages) [8] format. I believe these challenges may find elements of solutions in earlier works on scene description and I plan on keeping that activity as part of my future research.

---

[6] Web Component Standards `https://www.w3.org/standards/techs/components#w3c_all`

[7] Facebook's Instant Articles Project `https://instantarticles.fb.com/`

[8] Google's AMP Project`https://www.ampproject.org/`

Chapter 4

Multimedia Application Adaptation

Multimedia adaptation refers to the ability of a multimedia system to dynamically react to changing conditions [57]. These conditions can be related to the user. For instance a system may decide to offer different multimedia content to different users depending on their preferences or history. This is often referred to more specifically as content personalization. Some aspects of my work on content personalization were described in Chapter 3. The conditions against which a multimedia systems adapts can also be related to the user's device capabilities or to the network state. This is the focus of this chapter.

Multimedia adaptation to device capabilities or network state either relies on performing on-the-fly operations on the multimedia content to fit the given constraints (transcoding, transmoding, transformatting), or on selecting a suitable version of the content prepared in advance (e.g. selecting between a high resolution image and a low resolution one). Some properties of media content may impact the complexity of the adaptation operations. Media scalability [58], [59], [60], [61] is one of them. Media scalability can be defined as the property of a media content that permits processing even if only a part of it has been received. Scalability is also referred sometimes to as progressive enhancement[1], for example in the Web world. When a media content has this scalability property, adaptation can be as simple as not downloading a part of the stream or as truncating it.

Multimedia systems often use a client-server architecture. In such archi-

---

[1] Wikipedia page on progressive enhancement `https://en.wikipedia.org/wiki/Progressive_enhancement`

tecture, adaptation algorithms can be run at the server side or at the client side. In such architecture, the server does not need to be the source of the content, it may be an intermediate server or a proxy. Server-side adaptation allows running complex algorithms while the clients remain simple. But server side adaptation systems require information about the user (such as device characteristics, location, preferences), which may bring issues with user privacy. Server side systems also tend to not scale well when the number of users grows. On the contrary, client-side adaptation systems do not expose user private information and scale well with the number of users, but at the cost of increasing the computing requirements at the client side. Client-side adaptation systems, where the client has all the information to decide on what adaptation to perform, are also called adaptive or responsive systems, in particular in the Web world. MPEG-DASH [62], CSS Media Queries[2], or Responsive Images[3] techniques are all examples of client-side media adaptation approaches in use today.

Multimedia adaptation has been one of my research area since my PhD, as part of the ISIS and DANAE European projects, when I encountered the concept of media scalability for the first time, and then in different aspects, as presented in this chapter.

4.1 Adaptive Images

One of my works on adaptation is the one published at the User Centric Media Conference [28]. In this paper, together with Pr. H. Hellwagner and B. Reiterer from the University of Klagenfurt, in the course of the NoE INTERMEDIA, we were concerned with the visualization of large images on small displays. Today's trend is to produce alternative images and to deliver the most appropriate image depending on the device characteristics, as part of so-called responsive image frameworks. In the work that I present here, which was done prior to the appearance of responsive images[3], we took a different path.

The use case was to offer a way to view a large image with a small device, in the context of a museum visit. The main resource was a single image, such as a famous painting. We did not produce alternative versions of the image. To solve this problem, we proposed to offer a dynamic or interactive view of the content. For instance, when viewing a large web site on a mobile, browsers allow the user to zoom on the web site content. The zoom

---

[2] CSS Media Queries Standard `https://www.w3.org/TR/css3-mediaqueries/`

[3] W3C Responsive Images Community Group `https://responsiveimages.org/`

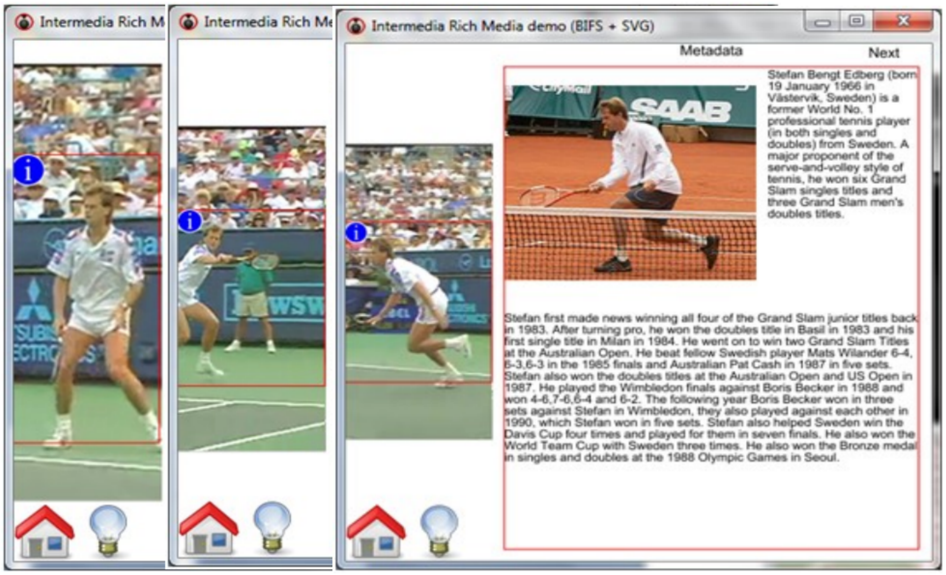
is typically guided by an analysis of the page to determine independent areas of interest, and the user is assisted in determining the right zoom factor to avoid imprecise pinch zoom and to only show one independent area when the user clicks on it. In this paper, we took a similar approach but taking into account animations capabilities. The idea was to generate an animation from the image. The purpose of the animation was to change camera parameters (translation and zoom factor) applied to the image to show all the different parts of the image at a resolution corresponding to the device. The animation was driven by some input text accompanying the image, i.e. a precise description of the painting. The result of the animation was a virtual camera motion. At the same time a text-to-speech engine was used to produce an audio track describing the image. In this paper, I contributed a method to generate the animation from camera positions and to generate the final synchronized audio/graphics content. One challenge was to enable such animated and synchronized rendering on constrained mobile devices. For this reason, I opted for the use of the SVG language, constrained as described in my PhD.

4.2 Multimedia Documents Adaptation

As my previous research focused on scene descriptions and multimedia documents, a first part of my research on adaptation was focused on adaptation of multimedia documents, in the context of the European project DANAE. Adaptation of multimedia documents cannot be performed simply by adapting each individual media element, as reported in [63]. If, for instance, the quality of a video is drastically reduced because the bandwidth is not sufficient to transmit it in high quality, the presentation containing this video should probably not display the video in full screen. Instead, other elements of the presentation such as text, graphics or images could be scaled. Similarly, if an element is removed because of the constraints of a device (eg. video codec not available), the entire presentation should be modified to provide equivalent content.

In DANAE, we proposed an end-to-end architecture for media adaptation. This architecture relied and extended the MPEG-21 standard [64]. We published a description of our work and in particular of the architecture of our system at the European Symposium on Mobile Media Delivery [47]. This work was then continued during the INTERMEDIA NoE. In that project, we worked on an improved adaptation system that also offered users a unified access to their adapted content, in the home network or through mobile networks. We published the resulting work at the Computer Graphics Conference [42] and in the Visual Computer Journal [40].

As an other example of my work on document adaptation, I presented during the Workshop on Social, Adaptive and Personalized Multimedia Interaction and Access [22] a client-side system capable of generating adaptive multimedia documents for the display of video content. In this system, depending on user action, either the video was shown in full screen or some additional images and descriptive text were shown next to the video. In that second view, the display of the video was adapted by showing only the region of interest of the video along side with the image and text elements, and with a different scene layout depending of the screen size, the aspect ratio of the region of interest and the quantity of side annotations. An example is shown in 4.1, which used alternative and fluid layouts, as deployed today in web responsive design systems.



Stefan Bengt Edberg (born 19 January 1966 in Västervik, Sweden) is a former World No. 1 professional tennis player (in both singles and doubles) from Sweden. A major proponent of the serve-and-volley style of tennis, he won six Grand Slam singles titles and three Grand Slam men's doubles titles.

Stefan first made news winning all four of the Grand Slam junior titles back in 1983. After turning pro, he won the doubles title in Basil in 1983 and his first single title in Milan in 1984. He went on to win two Grand Slam Titles at the Australian Open. He beat fellow Swedish player Mats Wilander 6-4, 6-3,6-3 in the 1985 finals and Australian Pat Cash in 1987 in five sets. Stefan also won the doubles titles at the Australian Open and US Open in 1987. He played the Wimbledon finals against Boris Becker in 1988 and won 4-6,7-6,6-4 and 6-2. The following year Boris Becker won in three sets against Stefan in Wimbledon, they also played against each other in 1990, which Stefan won in five sets. Stefan also helped Sweden win the Davis Cup four times and played for them in seven finals. He also won the World Team Cup with Sweden three times. He also won the Bronze medal in singles and doubles at the 1988 Olympic Games in Seoul.

Figure 4.1: Example of adaptive display of a video with a region of interest and associated annotations

More importantly, together with B. Pellan, as part of his PhD, we studied the issue of adapting entire multimedia documents by using the adaptation decisions taken at the media level. We used a server-side architecture as shown in Figure 4.2 inspired from the MPEG-21 architecture, within which we added a *Scene Adapter* component that jointly adapts the scene when the media elements are adapted.

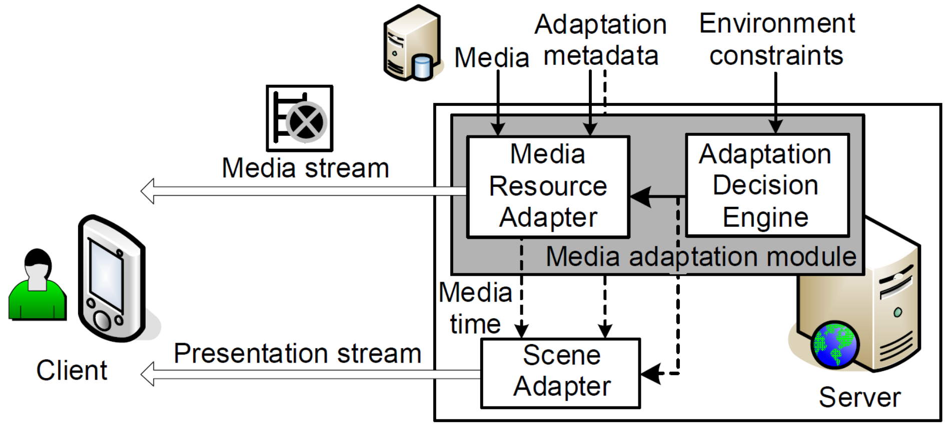


Figure 4.2: Architecture of a media-driven MPEG-21 Adaptation Engine, with the inclusion of Scene Adapter

An example of resulting adaptation is shown in Figure 4.3. When the video bandwidth is decreased, at a given point, the video element is replaced by a slideshow of some of its video frames, with animated transitions, and including with 3D effects. This work was published at the Workshop on Image Analysis for Multimedia Interactive Services [46].

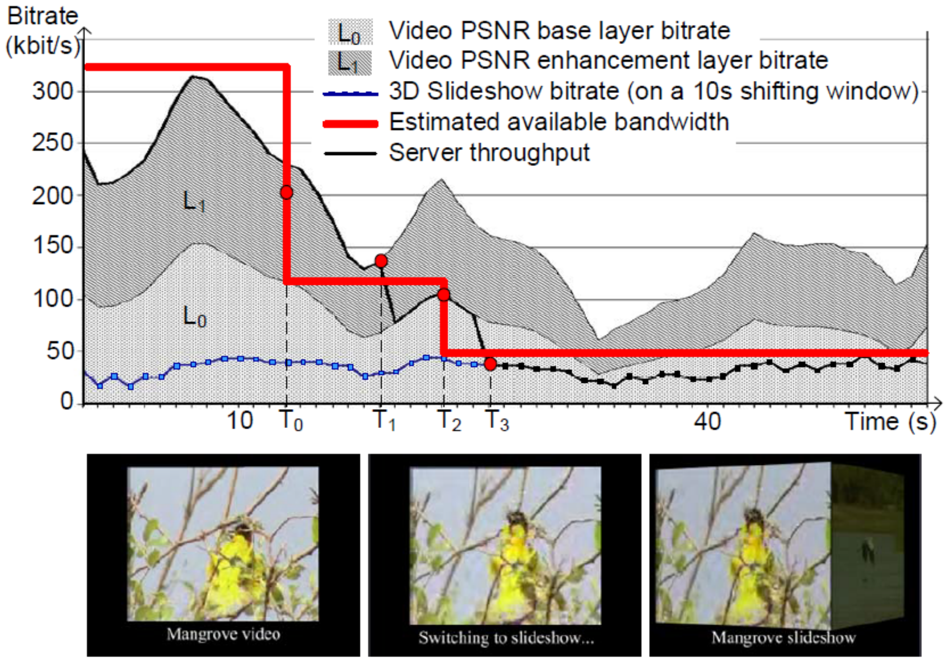
4.3 Broadcast-Friendly Adaptation using Scalable Multimedia Documents

Still with B. Pellan, as part of his PhD, we studied scene adaptation in broadcast environments, as reported here.

Adaptation in broadcast obviously differs from adaptation in Unicast. In Unicast, for instance in web environments, the return channel from the client to the server enables a constructive dialog between the client and the server. This dialog can be used to negotiate the best fitted content: either because the client sends its characteristics and lets the server determine or generate the most adapted content, e.g. using IETF Client Hints[4], or because the server provides a description of the alternative content and lets the client decide, e.g. in Responsive Images or DASH. In Broadcast, the same content is sent to all receivers. Hence, the adaptation approach which

---

[4] `http://httpwg.org/http-extensions/client-hints.html`



$[0 \ldots T_2]$ $[T_2 \ldots T_3]$ $[T_3 \ldots T_\infty]$

Figure 4.3: Example of combined scene and scalable media adaptation

consists in producing alternative versions of the same content and delivering them to all clients would not be efficient. Additionally, broadcast receivers are often simple processors. The adaptation logic must be simple. Complex operations, such as complex transcoding, on the media or the scene done at the client is not a possible choice. Media scalability offers an interesting compromise in broadcast because alternative versions of the same content can be delivered at a lower cost than simulcasting all versions by using an efficient encoding, and because the adaptation process is simple, as it usually consists in simply truncating a stream. However, scalability for scene descriptions had never been studied.

To study that concept, we first started by looking at the spatial aspects of scene descriptions. During the International Conference on Multimedia and Exhibition [41], we proposed and evaluated a broadcast-friendly spatial adaptation technique for scenes. In this paper, we proposed to deliver several scene layouts using Digital Multimedia Broadcasting DMB streams [65]. We encoded a base layout and layers made of MPEG-4 BIFS updates enabling to incrementally change the base layout as more updates were processed. An

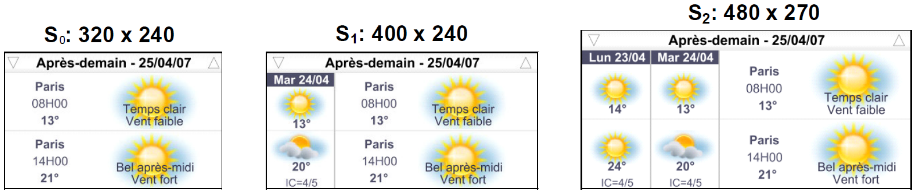


Figure 4.4: Example of enrichment of a scene based on screen dimensions using MPEG-4 BIFS updates

example is shown in figure 4.4. We demonstrated in this paper that BIFS updates offered an interesting compromise for adaptive layout compared to other scene description tools such as scene switching or interpolation.

Based on these results, we extended our work to the general concept of scalable multimedia documents, including the temporal and interactive aspects. We realized that multimedia documents structures actually contained differentiated elements, i.e. that some elements are more important than others. For instance, the elements indicating that a piece of text should be bold is not as important the text itself. As an other example, in a document like a YouTube page displaying a main video and related comments or videos, the related elements are less important than the main video. It seemed interesting to rely on this relative importance of elements to prepare a scalable version of multimedia documents, i.e. by defining scene layers and placing important elements in the base layer and less important elements in additional layers. A challenge in this approach is that document structures are not always clear. There are several concurrent structuring and the content of a document cannot easily be represented incrementally along a single axis. To simplify the adaptation process, we proposed to structure a document along 3 axes: spatial, temporal and interactive. We then proposed to build documents along each axis using a layered approach. A detailed description of the model has been presented at the ACM Symposium on Document Engineering [37], where some demonstrations of a player capable of processing such scalable documents were also presented in [39]. We concluded this work by studying the challenges related to the authoring of such documents. These results were published in Springer's Multimedia Tools and Applications journal [33]. As a side result of this authoring study, we investigated the summarization of such scalable documents and published our findings at the Workshop on Image Analysis for Multimedia Interactive Services [35].

4.4 Conclusion

With the heterogeneity of multimedia devices and the disparity of network accesses, adaptation of multimedia content is still a major topic. With the convergence between the media formats and the ubiquity of web browsers, the problem has evolved. The question of the adaptation of multimedia documents is now relevant from two different perspectives: from the HTML/CSS/JS one with technologies such as the *picture* element[5] or CSS Media Queries, and the other one from the media world, with adaptive streaming technologies. This latter point is the focus of the next chapter. In the future, I expect the issues on which I worked, such as document adaptation and media-document adaptation, more related to the former point, to become more and more relevant in the research context of HTML 5.

---

[5] Tutorial on the picture element `http://www.html5rocks.com/en/tutorials/responsive/picture-element/`

Chapter 5

HTTP Adaptive Streaming

Over the past few years, a new method for streaming content over the Internet has emerged. This method, called HTTP Streaming, is based on the use of the HTTP protocol. One of its goals is to overcome the deployment limitations of more traditional UDP-based streaming protocols. HTTP Streaming is designed to be firewall- and NAT-friendly and to scale well with number of users. Typical Internet infrastructures can be leveraged such as Content Delivery Networks (CDN). HTTP Adaptive Streaming is an evolution of HTTP Streaming, where the adaptive aspect is the result of the client choosing between several qualities advertised by the server through some manifest. A key aspect of HTTP Adaptive Streaming is that the content is split over time into chunks enabling clients to take multiple decisions over time about which quality it wants and therefore to adapt to varying network conditions. A lot of research has been done and is still on-going in this area. Among the major research topics, certainly the problem of finding the optimal adaptation algorithm, i.e. the one that produces the best quality of experience, is the topic that has received the most attention [67] [68] [69] [90] [91]. My research in this area has been carried in a slightly different direction, around several other aspects of the adaptive streaming: around extensions to the media formats used in adaptive streaming, namely on the ISO Base Media File Format (ISOBMFF); on the delivery of new types of streams besides audio and video, in particular subtitles, metadata, and other application data; and on improving the latency of the adaptive systems, in particular of the MPEG DASH standard [62]. This chapter presents these works.

5.1 Standardization activities

In the HTTP Adaptive Streaming ecosystem, among the proprietary solutions such as Apple HTTP Live Streaming (HLS) [49] or Microsoft Smooth Streaming [66], the MPEG DASH standard emerged as a common format between different industrial companies and consortia for representing the manifest format and for constraining the formats of the media chunks. As you will see, my research in the HTTP Adaptive Streaming field is heavily tied with the DASH standard. I proposed more than 30 contributions to this standard and I have been co-editor of the 2nd amendment to MPEG-DASH, editing in particular the MPEG DASH Spatial Relationshipt Description (SRD) feature. As a consequence, our open source software was the first, world-wide, to propose an implementation of the DASH standard, both to generate DASH content and to play it. We also demonstrated a first open-source tool to generate live DASH content at the IEEE Workshop on Multimedia Signal Processing [12]. Finally, together with the University of Klagenfurt, we also offered a data set based on MPEG-DASH for researchers to test their adaptation algorithms. This data set was published at the ACM Multimedia Systems Conference [15].

This activity also led me to contribute as well to various Web standards. In particular, I have participated in the design of the standards that drive how web browsers consume adaptive media data. The relevant standards are HTML 5, in particular its Media Source Extension[1] and the mapping of media resources to HTML Tracks[2]; and the specifications of the Timed Text Working Group[3].

5.2 Adaptive Streaming of Text Content

Amongst the media assets used in multimedia applications, text streams are used for subtitles, closed captioning or in general for accessibility reasons. Text streams typically have a very low bit rate compared to audio or video streams, are split into units of variable durations, and have content that may still require complex processing at the client-side, such as animations or font processing. Due to these characteristics, some systems choose to convert text streams to image streams to reduce the associated client-side complexity. In that case, the adaptive streaming of text streams is similar

---

[1] Media Source Standard `https://www.w3.org/TR/media-source/`

[2] Draft of mapping of media tracks to HTML `https://dev.w3.org/html5/html-sourcing-inband-tracks/`

[3] W3C Timed Text Working Group`https://www.w3.org/AudioVideo/TT/`

to the adaptive streaming of video streams. This is however not satisfactory for accessibility reasons. In my research, I concentrated my activities on text-based formats.

A plethora of subtitle formats exists, from formats used in the broadcast world such as DVB Subtitles [70], to formats used in the mobile world like 3GPP Timed Text [71], to formats used in the open-source community such as SRT[4]. With the rise of HTML5 and the ubiquitous use of browsers, two additional formats have emerged: Timed Text Text Markup Language (TTML[5]) and WebVTT[6]. TTML is a markup language, based on XML, designed initially as an authoring format, with all the features required for authoring, and then used as a distribution format. WebVTT is a format, largely inspired from SRT, designed to integrate well with the rest of the Web Platform, in particular with CSS and JavaScript. My research in this area focused on the storage, streaming, and adaptive streaming of text streams, particularly using WebVTT, TTML, the MPEG ISOBMFF and DASH technologies.

As for video and audio streaming, there are two important categories of text streaming systems: on-demand and live. In the on-demand scenario, since the entire video and subtitles are available when the playback request is made, the client will typically download the whole WebVTT/TTML file and start processing it synchronously with the video. In practice, the download and processing of the subtitle and video data are done in parallel. This is the major Web-related use case for which WebVTT/TTML were designed.

The live streaming scenario is more challenging. For live HTTP adaptive streaming scenarios, the video is delivered in chunks. For these scenarios, in order to avoid delaying the rendering of video, the text content also has to be delivered in chunks. Each chunk has to be delivered soon after it is produced to reduce the latency of the system. This raises the questions of how to chunk a subtitle file and of how to signal properties of chunks such as random access, in an efficient way. My research in this area tried to provide answers to these questions, and is summarized here.

TTML and WebVTT are file-based formats, typically not progressively processed by browsers or players, which makes them bad candidate for streaming. One approach to solve this problem is to deliver text content in small independent files processed in sequence, but still not progressively. That is the approach chosen by Apple to stream WebVTT content in the

---

[4] Wikipedia page on SubRip Text format `https://en.wikipedia.org/wiki/SubRip`

[5] TTML Standard `https://en.wikipedia.org/wiki/SubRip`

[6] WebVTT standard `https://www.w3.org/TR/webvtt1/`

[49] specification[7]. This is also the approach we used in an early demonstration of live subtitle streaming in browsers, as shown at the ACM Multimedia Systems Conference [14]. This simple approach does not however facilitate the storage of the subtitle content together with the video or audio content because of the multiplicity of files. We thus proposed a different approach, based on the storage of WebVTT content and TTML content in MP4 files. This lead to the standardization of MPEG-4 Part 30, for which I was co-editor and produced more than 10 contributions.

One challenge in this storage is the processing of *overlapping cues*. In subtitle content, it is often the case that consecutive text chunks, also called text cues, either are separated by a gap or to the contrary overlap. The first case is for example when a movie has silence. The second case happens for instance when two characters in a movie speak at the same time. This is a unique feature compared to other traditional media types and this is not compatible with typical storage formats such as the ISOBMFF. In that format, streams are cut into access units (AU) that have distinct start times and the end of an AU is the start of the next one. Cutting a stream into AU for storage in an ISOBMFF file is a similar process as cutting a stream into segments of an adaptive streaming system, but with a finer granularity. Additionally, seeking into a file or into an adaptive stream requires signaling of random access point. To address these issues, I recommended several approaches. The first one[8] consists in an algorithm to produce non-overlapping cues from overlapping ones as depicted in 5.1 and 5.2. The second approach consists in producing adaptive streaming segments with duplicate text chunks across segments, but with the duplication signaled as redundant, using a specific feature of the ISOBMFF. Both approaches are now used in industrial deployments.

5.3 HTTP Streaming Latency

In today's networks, in particular in mobile networks, the available bandwidth can vary drastically. As a solution to this problem, HTTP Adaptive Streaming techniques typically use large buffers to smooth the bandwidth variation and to prevent clients' buffer underflows, adding to the latency of the system, including live systems [73]. A common industrial practice is to buffer as much as 30 seconds and some on-demand systems even recommend up to 240 seconds [72].

---

[7] https://tools.ietf.org/html/draft-pantos-http-live-streaming-18

[8] https://concolato.wp.mines-telecom.fr/2012/09/12/webvtt-streaming/

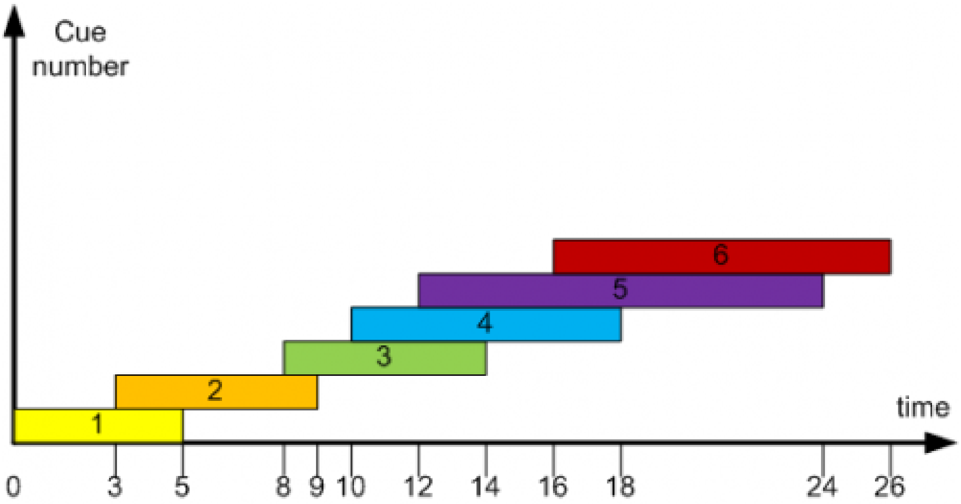


Figure 5.1: WebVTT overlapping cues before split: random accessing at a time T requires data from time T and other time ranges

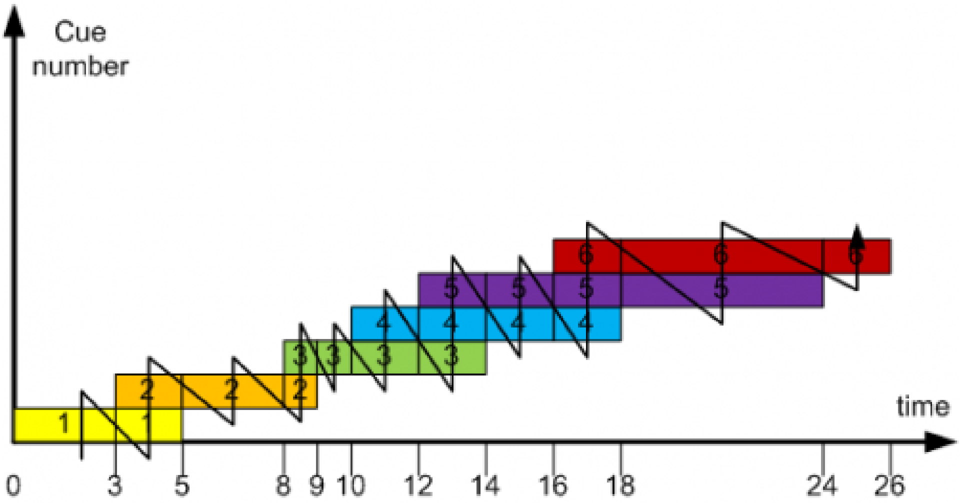


Figure 5.2: Split WebVTT overlapping cues stored in ISOBMFF Files: random accessing at a time T only requires data from that time

Streaming over HTTP is not entirely new. Earlier HTTP-based solutions existed that offered reduced latency compared to the initial deployments of HTTP adaptive streaming. Using those techniques, web radios have been deployed for a long time using techniques such as ShoutCast[9] or IceCast[10]. Web radios stream their content on HTTP as part of a never-ending file download. In that approach, the HTTP protocol is slightly modified as the Content-Length is advertised to be ”0” such that the client is informed that the file is in fact a stream. As opposed to HTTP (Adaptive) Streaming, the client issues only one HTTP request. This modification has a strong impact on caches. Two clients joining at different times will issue the same request (with the same parameters) but will receive (different) live content. It is therefore not possible to cache such request with general purpose caching techniques.

Several existing approaches try to reduce this latency: some of them focus on new protocols such as the push feature of HTTP/2 [78] [89], some others reuse existing HTTP/TCP tools [88]. Inspired by this example of Web radios, together with J. Le Feuvre and N. Bouzakaria, as part of her PhD work, we investigated a backwards-compatible approach with the use of HTTP Chunk-transfer encoding to reduce the streaming latency while preserving cacheability. If one tries to reduce the latency by reducing the segment length, one obvious cost is an increase in the number of HTTP requests, with its consequences on upstream traffic and on server load. In our approach, we kept the notion of segment unmodified, i.e. as a request unit, and used the notion of fragment as an individually deliverable part of a segment that is not individually requested. This however increases the packaging overhead. We measured this overhead when using segments with typical duration (1 or 2 seconds) and with very small durations, with fragments as short as one video frame. Unsurprisingly, the overhead proved to be inversely proportional to the fragment duration. However, we noted that the overhead remained perfectly acceptable, as low as 2-3%, for videos with high bit rates. We then combined the use of fragments and the use of HTTP/1.1 chunked-transfer encoding to enable the reduce the latency of DASH streaming system. With this approach, we showed that the latency is not anymore in the order of the segment duration but of the fragment duration. This work was published at the QoE session for Over-The-Top Multimedia Services at the Conference on Information, Intelligence, Systems and Applications [10] and is being used in several industrial deployments.

One other important aspect of web streaming, related to latency, is the end-to-end delay, i.e. the delay between the capture of a stream and its viewing by a user. Typical legacy broadcast systems induce a delay of around

---

[9] `https://en.wikipedia.org/wiki/SHOUTcast`

[10] `https://en.wikipedia.org/wiki/Icecast`

7 seconds (in France using terrestrial TV). Bi-directional systems tolerate even fewer delay as indicated in [85]. As opposed to that, typical adaptive HTTP streaming deployments induce a delay of more than 30 seconds. This reduces the adoption of AHS solutions and limits it to on demand streaming only. One way to reduce this delay in AHS is to reduce the initial phase of AHS solutions where the client downloads the manifest and initialization segments. As part of N. Bouzakaria's PhD work, we published a study which shows that the current compromise in which the clients download only the necessary initialization segments but using multiple HTTP requests was not the best. We showed that downloading all the initialization segments and the manifest in a single request proved to be faster on HTTP/1.1-based delivery and to a lesser extent in HTTP/2. This work was published at the Workshop on Multimedia Signal Processing and received the top 10% best paper award [3] and is being integrated in the next amendment to MPEG-DASH.

Following these studies, we also improved our open source software to experiment and demonstrate the use of low latency AHS streaming. We demonstrated its use in two scenarios: in hybrid delivery mixing broadband and broadcast and in bidirectional video communications. These demonstrations were made at the ACM Multimedia Conference [4].

5.4 Adaptive Streaming of Tiled Videos

With the development of Ultra-High Definition (UHD) videos, the bandwidth required to deliver such videos is exponentially growing. Despite the development of new video codecs such as the High Efficiency Video Codec (HEVC) [74], it is still difficult to deliver such videos in bandwidth-limited networks. This impacts a large range of applications: from video surveillance, to immersive reality, to medical image ... Different researchers have studied the use of multiple smaller resolutions videos, often called tiles, to solve this problem [75][76][77][82]. Their solutions often require dedicated clients or servers in systems that do not scale well with the number of users.

Interestingly, the HEVC codec defines a way to encode tiles of a given video independently from each other in a single bitstream. Based on this technology, together with Canon Research Centre France, I recently started studying the storage and delivery of such tiled images or videos. In particular, I worked on the storage of HEVC tiled images and videos in the HEVC file format, which is based on the ISOBMFF. This led to the submission of 7 contributions to MPEG and several patents. I studied also how the

MPEG-DASH standard could be extended to enable the adaptive streaming of tiled videos, and in particular of HEVC tiled videos. This led to several contributions to MPEG DASH, and to the publication of the Spatial Relationship Description amendment, for which I was co-editor. Several publications were made on this topic and presented at the ACM Multimedia Systems conference [1] and [2] and a journal paper is under review.

5.5 Conclusion

HTTP Adaptive Streaming technologies are now mature: from the industrial perspective, and from the academic one. In the industry, the deployment of multiple technologies, similar but slightly different, creates market fragmentation. Efforts are being made by many consortia to make these technologies converge. In particular, the Common Media Application Format is being drafted within a new group at MPEG for which I have been appointed co-chair. From an academic perspective, many new research problems arise. Different optimizations of these adaptive streaming technologies are required. In the mobile world, the ability for the networks to assist clients in selecting the right quality is a key problem on which I intend to work with Huawei technologies via a CIFRE PhD. In the Web world, the reduction of the load on CDN by the using of peer-to-peer networks is also an interesting aspect on which I will work via a CIFRE PhD with StreamRoot. Finally, the problem of designing algorithms to choose the quality of videos in virtual reality adaptive streaming systems is also a domain to which I would like to contribute.

Chapter 6

Multimedia Application Delivery

Delivering a multimedia application can be seen as the simple task of delivering a set of files. If we consider Web Applications, made of HTML, CSS, JS or image resources, delivering such applications can simply consist in delivering the entry resource, i.e. the HTML page, and then recursively delivering all linked resources. Such simple delivery approach works but has limitations. One limitation is latency. In some cases, if the bandwidth permits, one might want to receive some first-level resources in parallel, e.g. HTML, CSS and JS, to avoid latency in the final rendering of the application. More generally, the limitation is the lack of consideration for the dependencies between resources. Some resources may require that another resource be loaded before actually being useful. A lot of research is currently on-going in this area. The use of the new HTTP/2 features, such as resource multiplexing, resource push or prioritization, is being investigated. Additional problems also arise when considering applications that contain audio/video content. Depending on the relationship between the application content and the audio/video content, specific delivery mechanisms are required. For instance, in some applications, the content of the application needs to be tightly synchronized with the audio/video. In the past years, I studied various of these aspects regarding the delivery of multimedia applications. This research is presented in this chapter.

6.1 Document Streaming

As a first work in this area, I presented at the ACM Symposium on Document Engineering [45] a method to deliver long running animated SVG content. As opposed to existing approaches which consisted in a simple progressive download and playback, this method proposed to fragment an application document into small document fragments, to be delivered at specific given times. Such approach is applicable when it is possible to structure a document along the time axis (possibly external as reported in [86]), rather than along the spatial axis, in which case level-of-details streaming [79] or specific graphics adaptive streaming [81] approaches are preferable. In that paper, this approach worked well for long-running SVG cartoons, and even more useful when the cartoon was synchronized with audio content as it allowed streaming synchronously the graphics and the audio content. In that approach, we also proposed to mark some fragments as not relying other fragments. Both of these functions, fragmentation and signaling of random access fragments, enable transforming SVG documents into SVG streams. In turn, this offers additional streaming capabilities such as live streaming or seeking. Finally, in that paper, we also showed that when coupled with a proposed declarative mechanism for garbage collection of unused SVG elements, this method proved to drastically reduce the memory consumption peak rate.

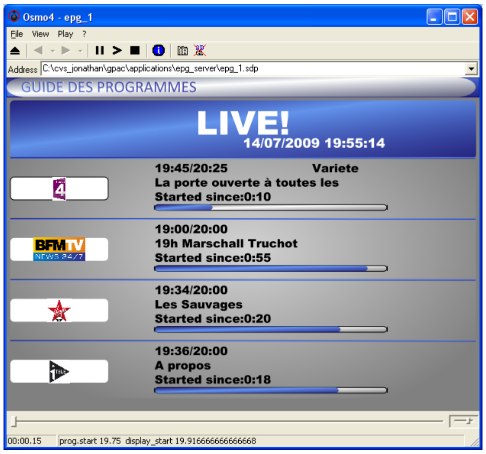
This work was also presented at the Graphical Web Conference [13], and led to the creation of a new part of the SVG specification, called SVG 2 Streaming Module[1], for which I am the editor. Also as a follow-up, together with E. Potetsianakis, we applied this work to the streaming of other kinds of web data and presented the associated results for XML data at the XML Prague Conference [8], and for 3D graphics at the Web3D Symposium [7].

6.2 Electronic Program Guide Streaming

The previous concept of document streaming was also reused for the delivery of Electronic Program Guides. This work was done in the context of the PINGO project. The goal of that project was to provide a home network gateway capable of relaying outdoor DVB-H signal into indoor WiFi signals. The mobile signal contained simple audio and video streams as well as additional information such as Electronic Program Guide (EPG) information. All data had to be relayed. My contribution was on the relaying of the EPG

---

[1] SVG 2 Streaming Module `https://dev.w3.org/SVG/modules/streaming/`



Address C:\cvs_jonathan\gpac\applications\epg_server\epg_1.sdp

14/07/2009 19:55:14

19:45/20:25 Variete
La porte ouverte à toutes les
Started since:0:10

19:00/20:00
19h Marschall Truchot
Started since:0:55

19:34/20:00
Les Sauvages
Started since:0:20

19:36/20:00
A propos
Started since:0:18

00:00.15 prog.start 19.75 display_start 19.916666666666668

Figure 6.1: Example of Streamable Electronic Program Guide

part. In the project, EPG data was received in the form of MPEG-2 Event Information Table (EIT), following the *carousel* delivery method. The target devices in the home network were mobile phones and tablets with typical web capabilities. We decided to use the SVG format to enable those devices to display the transcoded EPG data and to generate SVG data in such a way that the rendering would adapt on each client device to its characteristics, following work presented in Chapter 4. Since the EPG data was received from the broadcast, i.e. in a push mode, we decided to deliver the data to the client devices using the push mode. The document streaming approach defined earlier was adopted. The format of the stream, however, was made compliant to 3GPP DIMS[2]. This work was published at the EuroITV conference [34] and also presented during the SVG Open Conference [24]. An example of streamed EPG is given in Figure 6.1.

Interestingly, in that work, we generalized the approach and described a system which proposed to separate the delivery of raw data (in this case the EPG data) from the delivery of presentation instructions (in this case the SVG data). This paper presented a comparison of the methods delivering on only metadata to the client versus methods delivering only presentation data as illustrated in Figure 6.2. I think that this classification is relevant today for delivering metadata to devices. For instance, in scenarios using MPEG-DASH, this means chosing between delivering metadata representations or presentation-data representations.

---

[2] 3GPP DIMS Standard `http://www.3gpp.org/DynaReport/26142.htm`

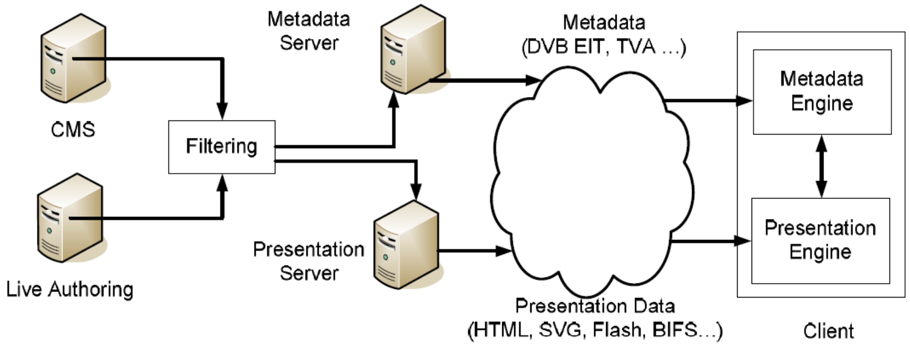


Figure 6.2: Architecture for the delivery and rendering of presentation data and metadata

6.3 Application broadcasting

Based on this work, we tried to derive a model for the delivery of interactive multimedia applications over broadcast channels. In this model, we proposed to distinguish in an interactive application, the elements, called base elements, which need to be carrouselled periodically with a short period to enable fast tuning, from the elements which should either be carrouselled with a long period or delivered as a single update. This model was presented at the Workshop on Mobile Video Delivery [23].

6.4 Application delivery over MPEG-DASH

As a follow-up of the previous works, at the ACM Multimedia Systems Conference, we presented an approach for the delivery of interactive multimedia applications over DASH [21]. In that work, we considered MPEG-4 BIFS applications. We first described the need for providing ways to identify parts of a DASH session for track selection from within interactive applications. This led to a modification of the MPEG-DASH standard and the definition of URL fragment identifiers for DASH content. We also presented how Rich Media services could be carried in a DASH session along with audio and video data to ensure tight synchronization between the media data and the interactive service, in particular how to build a smart, bandwidth-friendly data carousel of interactive services and meta-data in a DASH session.

6.5 Application delivery over ISOBMFF

Finally, MPEG-4 BIFS not being very popular, I also studied the general carriage of Web applications in the ISOBMFF. In particular, I studied how ISOBMFF items could be used to store out-of-band application. The use of such items to propose enriched audio/video presentations was presented at the Graphical Web conference [6] and the proposed modifications to the ISOBMFF are now part of the standard. Also as a consequence of this work I became the editor of the MPEG report on Interactivity in MPEG File Formats. This document is intended to survey all available options to carry interactive data and more generally metadata in the ISOBMFF. It is intended to serve as documentation for other standardization bodies.

6.6 Hybrid Multimedia Delivery

Hybrid multimedia delivery refers to the use of multiple networks or network types to deliver multiple pieces of multimedia content, which possibly have to be played synchronously. In a paper presented at the ACM Multimedia Systems conference [17], we introduced the general concept and challenges of such delivery. In general, hybrid delivery requires solving two issues: first, determining which content from the second network matches the content from the first network; and synchronizing the two pieces of content. In our paper, we surveyed different approaches for providing the information to link the different media streams. We called this process *bootstrapping*. We indicated that the bootstrap information could be at different layers of the delivery stack: in the protocol itself (e.g. in a MPEG-2 TS broadcast), in the application content (e.g. in the HTML5 page) or in-between (e.g. in a DASH MPD). We reviewed the synchronization issue too, indicating that it is usually resolved by using a common clock and presented different solutions. We highlighted the difficulty of choosing such clock, as depending on the given network type, the clock may not be sufficiently accurate or reliable.

One particular declination of hybrid delivery is the one mixing a broadcast network and a broadband network. This combination is interesting because it leverages the strengths of both types of networks. Broadcast networks are efficient at delivering the same content to a large number of users, while broadband networks with point-to-point connections can deliver personalized content to individual users.

In [17], we studied the use of hybrid delivery of multimedia content for

Digital Radio content, in the context of the HybRadio project. The goal of this project was to design technologies to enable the delivery of visual digital radio content, i.e. digital audio content enriched with a visual component (images, text or graphics). One of the main constraints faced by radio broadcasters was that the deployment of a new broadcast network was considered too costly and that the FM analog audio quality was deemed too good. The FM network had to be reused. Therefore technologies such as Digital Audio Broadcast (DAB) or Digital Multimedia Broadcast (DMB) initially envisaged could not be reused as is. In that study, we started envisaging hybrid delivery with analog audio content delivered over the FM network, and additional digital visual content delivered over an IP network, either 3G or WiFi. In this work, we suggested to use a common clock in both contents, delivered without relying on existing broadcast technologies for delivery time, as those were not reliable enough. We validated the concept through an implementation within GPAC of a module capable of synchronizing streams from multiple networks at the same time.

Following this work, we presented this activity to MPEG which led to the standardization of the MPEG-2 Timeline and External Media Information (TEMI) amendment, i.e. mechanisms enabling the localization and synchronization of content delivered outside of an MPEG-2 Transport Stream. This work has been extended by many other researchers and other standards have been also deployed such as DVB Companion Screens and Streams (DVB-CSS). To demonstrate this activity, we presented, at the ACM Multimedia Conference [4], an implementation within GPAC of a player capable of combining broadband MPEG DASH content with Broadcast MPEG-2 MPEG-2 TS content using MPEG-2 TEMI.

6.7 Application delivery over Hybrid networks

Hybrid delivery is relevant not only at the media level, but also at the application level, and in particular in Web applications. An author could decide to use two independent media streams in an application and to have them synchronized. For instance, a Twitter feed could be linked to a video stream during a soccer game. The concept of hybrid delivery is at the foundation of the Hybrid broadcast broadband Television (HbbTV) standard, which is related to the work on multiscreen presented earlier.

The HbbTV standard introduced the concept of Hybrid Broadcast Broadband application. In an HbbTV application, part of the content is received from the broadcast channel, typically audio/video content, and part of it is received via the broadband channel, typically a Web application composed of HMTML, CSS, JavaScript, images, etc. An interesting problem is to transform an HbbTV application from a live service to an on-demand service. We studied this problem and presented our results at the EuroITV conference [19]. In particular we presented an analysis from three different perspectives. We first looked at the management of resources, then at transport and carousel issues. Finally, we look at the modifications that need to be performed at the application level to ensure its correct playback.

6.8 Conclusion

Multimedia application delivery is a challenging task as many resources have to be delivered that have different characteristics. Some are streams that are produced continuously, that may tolerate errors. Some are simple files that have to be delivered first, entirely or progressively, but without errors. Most of them have constraints on synchronization. In my past research activities, I have studied a large set of aspects related to multimedia application delivery, in different contexts. I think that gives me a particular understanding of this area. With the ubiquitous use of HTML5 to develop multimedia applications, the availability of the IP network and the beginning of the deployment of HTTP/2, we are already seeing the deprecation of old delivery practices such as the use of resource concatenation or the use of domain sharding [80]. I think we will see in the future an important evolution of the delivery of multimedia applications. However, there is still no study on the optimal delivery mechanism, for instance in terms of application latency. I believe that this would be an interesting topic of research for the future. In terms of standardization, the definition of application programming interfaces for browsers would also be a important task to offer advanced delivery options.

Chapter 7

Conclusion and perspectives

In this report, I have presented my research work since the end of my PhD. From this presentation, it can be seen that my research activities span two different fields. An significant part of my research activities covers the field of Computer Science. This applies to my earlier works on Scene Descriptions, on document engineering and to my more recent works on Computer Graphics, on interactive applications, and on Web standards. Networking is the second field of research to which I participate, with works on streaming technologies or on content delivery technologies in general. These two fields have met in my works on multimedia adaptation or multimedia application delivery and will likely meet again in my future work. I think this double expertise is a strength. It gives me a particular view of the domain of multimedia. For instance, few experts in Web technologies know about networking or streaming technologies, and vice-versa. It gives me also the opportunity to bridge the two communities, with the drawback that my work may fully interest only a restricted set of researchers.

My research perspectives will naturally cover these two fields. From a computer science point of view, I intend to continue working on Web standards. In particular, I believe multimedia applications could benefit from accessing more types of streaming data, and not only audio and video streams. Metadata streams such as those produced by the sensors of our mobile devices could enable interesting new usages. This requires an evolution of current Web programming interfaces. I intend to work on this aspect with the starting PhD of E. Potetsianakis. I would like also to progress the streaming graphics capabilities of Web, with the SVG Streaming Module.

From a networking point of view, the emergence of new protocols will provide new challenges and opportunities to improve adaptive streaming technologies. For instance, the HTTP/2 standard recently published opens the way for new possibilities such as the ability to multiplex data in a single connection, possibly weighted, offering interesting options when adaptively streaming multiple videos, such as in tiled panoramic or 360-degree video streaming. I have already started studying these aspects with some of my industrial partners. Another interesting emerging protocol is WebRTC. This peer-to-peer protocol, available in many Internet browsers today, enables effectively using hybrid delivery of content within the browser: from peer-to-peer channels and from traditional channels, using content delivery networks. In terms of adaptive streaming, this offers new possibilities for bitrate adaptation algorithms. I already supervised a Master student (L. Bracco) on this aspect and I plan to further study such algorithms, and already have an open PhD position with a start-up of the field. Finally, other protocols such as QUIC opens up the way for large-scale Internet adaptive streaming over the non-reliable UDP protocol. This is an interesting evolution which may see application of my works on reducing the adaptive streaming latency.

Glossary

AAC MPEG Advanced Audio Coding. 6

BIFS BInary Format for Scenes. 5, 7, 8, 14

CSS Cascading Style Sheets. 5

DAB Digital Audio Broadcasting. 40

DANAE Dynamic and distributed Adaptation of scalable multimedia coNtent in a context-Aware Environment. 20, 21

DASH Dynamic Adaptive Streaming over HTTP. 6, 40

DMB Digital Multimedia Broadcasting. 24, 40

DVB Digital Video Broadcasting. 29

H.264—AVC MPEG Advanced Video Coding a.k.a ITU-T H.264. 6

HTML Hypertext Markup Language. 3, 5, 7

HTML5 Hypertext Markup Language. 5, 6, 18

INTERMEDIA Interactive Media with Personal Networked Devices. 12

IP IETF Internet Protocol. 6

ISIS Intelligent Scalability for Interoperable Services. 20

ISOBMFF ISO Base Media File Format. 3, 6

LASeR Lightweight Application Scene Representation. 14

MPEG Moving Pictures Experts Group. 3, 5, 16, 40

MPEG-2 TS MPEG-2 Transport Stream. 40

NoE Network of Excellence. 12, 20

RoI Region of Interest. 12, 14

SMIL Synchronized Multimedia Integration Language. 5, 7

SRT Sub Rip Text. 29

SVG Scalable Vector Graphics. 3, 5, 7–10

TEMI Timeline and External Media Information. 40

TTML Timed Text Markup Language. 29

VRML Virtual Reality Modeling Language. 8

W3C World Wide Web Consortium. 3, 5

WebVTT Web Video Timed Text. 29

X3D Extensible 3D format. 8

Bibliography

[1] O. Niamut, E. Thomas, D. Mico, C. Concolato, F. Denoual, and S. Y. Lim. “MPEG DASH SRD - Spatial Relationship Description”. In: *ACM MMSys*. Klagenfurt, Austria, May 2016.

[2] J. Le Feuvre and C. Concolato. “Tiled-based Adaptive Streaming using MPEG-DASH”. In: *ACM MMSys*. Klagenfurt, Austria, May 2016.

[3] N. Bouzakaria, C. Concolato, and J. Le Feuvre. “Fast DASH Bootstrap”. In: *MMSP*. Xiamen, China, Oct. 2015.

[4] J. Le Feuvre, C. Concolato, N. Bouzakaria, and V. T. Nguyen. “MPEG-DASH for Low Latency and Hybrid Streaming Services”. In: *ACM MM*. Brisbane, Australia, Oct. 2015.

[5] M. Sarkis, C. Concolato, and J. C. Dufourd. “MSoS: A Multi-Screen-Oriented Web Page Segmentation Approach”. In: *DocEng 2015*. Lausanne, Switzerland, Sept. 2015.

[6] C. Concolato and E. Potetsianakis. “Packaging web graphics and media for easier internet delivery”. In: *The Graphical Web*. Pittsburg, USA, Sept. 2015.

[7] C. Concolato, J. Le Feuvre, and E. Potetsianakis. “Synchronized Delivery of 3D Scenes with Audio and Video”. In: *Proceedings of the Twentieth International ACM Conference on 3D Web Technologies*. ACM, June 2015.

[8] C. Concolato and E. Potetsianakis. “In-Browser XML Document Streaming”. In: *XML Prague 2015*. Prague, Czech Republic, Feb. 2015, pp. 197–205.

[9] M. Sarkis, C. Concolato, and J. C. Dufourd. “The Virtual Splitter: Refactoring Web Applications for the Multiscreen Environment”. In: *ACM DocEng-2014*. Fort Collins, Colorado, USA, Sept. 2014, pp. 139–142.

[10] N. Bouzakaria, C. Concolato, and J. Le Feuvre. “Overhead and Performance of Low Latency Live Streaming using MPEG-DASH”. In: *IISA QoE for Over-The-Top Multimedia Services*. Chania, Greece, July 2014, pp. 92–97.

[11] P. Ilbery, L. Kendall, C. Concolato, and M. McCosker. “Biharmonic diffusion curve images from boundary elements”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2013)* 32.6 (Dec. 2013), 219:1–12.

[12] A. Shafiei, C. Concolato, and J. Le Feuvre. “DashCast, A Live DASH Streaming Server”. In: *MMSP*. Pula, Italia, Oct. 2013.

[13] C. Concolato. “Streaming SVG animations on the Web”. In: *The Graphical Web*. San Francisco, USA, Oct. 2013.

[14] C. Concolato and J. Le Feuvre. “Live HTTP Streaming of Video and Subtitles within a Browser”. In: *ACM MMSys*. Oslo, Norway, Feb. 2013.

[15] S. Lederer, Ch. Mueller, Ch. Timmerer, C. Concolato, J. Le Feuvre, and K. Fliegel. “Distributed DASH Dataset”. In: *ACM MMSys*. Oslo, Norway, Feb. 2013.

[16] Y. Lim, C. Concolato, J. Le Feuvre, and K. Kim. “The MPEG Representation of Digital Media”. In: Springer, 2012. Chap. 10 : MPEG Multimedia Scene Representation, pp. 177–202.

[17] C. Concolato, S. Thomas, R. Bouqueau, and J. Le Feuvre. “Synchronized delivery of multimedia content over uncoordinated broadcast broadband networks”. In: *MMSys*. Vol. Proceedings of the 3rd Multimedia Systems Conference. Chapel Hill, NC, USA: ACM New York, NY, USA ©2012, Feb. 2012, pp. 227–232.

[18] M. Gowing et al. “Enhanced Visualisation of Dance Performance from Automatically Synchronised Multimodal Recordings”. In: *ACM Multimedia*. Scottsdale, Arizona, USA, Nov. 2011.

[19] J. C. Dufourd, S. Thomas, and C. Concolato. “Recording and Delivery of HbbTV Applications”. In: *EuroITV*. Lisbonne, Portugal: Springer, June 2011, pp. 51–54.

[20] C. Concolato, J. C. Dufourd, J. Le Feuvre, K. Park, and J. Song. “Communicating and migratable interactive multimedia documents”. In: *Multimedia Tools and Applications* (May 2011).

[21] C. Concolato, J. Le Feuvre, and R. Bouqueau. “Usages of DASH for Rich Media Services”. In: *Multimedia Systems Conference*. San Jose, USA, Feb. 2011.

[22] C. Concolato. “Adaptive Video and Metadata Display using Multimedia Documents”. In: *International Workshop on Social, Adaptive and Personalized Multimedia Interaction and Access*. Florence, Italy: ACM, Oct. 2010.

[23] C. Concolato and J.-C. Moissinac. “A Model for the Delivery of Interactive Applications over Broadcast Channels”. In: *Workshop on Mobile Video Delivery (Movid 2010)*. Florence, Italy: ACM, Oct. 2010.

[24] C. Concolato and J. Sillan. “SVG Electronic Program Guides”. In: *SVG Open 2010*. Paris, France, Sept. 2010.

[25] S. De Bruyne, P. Hosten, C. Concolato, M. Asbach, J. De Cock, M. Unger, J. Le Feuvre, and R. Van de Walle. “Annotation based personalized adaptation and presentation of videos for mobile applications”. In: *Multimedia Tools and Applications* 50 (July 2010), pp. 1–25.

[26] J.-C. Moissinac, C. Concolato, and J. Le Feuvre. “SuperPath: a necessary primitive for vector graphics formats”. In: *MMEDIA 2010*. Athen, Grèce, June 2010.

[27] A. Lorenz, M. Jentsch, C. Concolato, and E. Rukzio. “A Formative Analysis of Mobile Devices and Gestures to Control a Multimedia Application from the Distance”. In: *MELECON*. Malte, Apr. 2010, pp. 796 –801.

[28] B. Reiterer, C. Concolato, and H. Hellwagner. “Natural-Language-based Conversion of Images to Mobile Multimedia Experiences”. In: *International Conference on User Centric Media*. Vol. 40. Venezia, Italy: Springer, Dec. 2009, pp. 87–90.

[29] J. C. Dufourd, C. Concolato, and J. Le Feuvre. “SVG Communicating Widgets”. In: *SVG Open*. Mountain View, CA, USA, Oct. 2009.

[30] S. De Bruyne, J. De Cock, R. Van de Walle, P. Hosten, M. Asbach, M. Wien, and C. Concolato. “Personalized adaptation and presentation of annotated videos for mobile applications”. In: *5th international ICST Mobile Multimedia Communications Conference*. London, United Kingdom, Sept. 2009.

[31] C. Concolato, J. Le Feuvre, and J. C. Dufourd. “Declarative Interfaces for Dynamic Widgets Communications”. In: *Document Engineering*. Munich, Germany: ACM New York, NY, USA, Sept. 2009, pp. 241–244.

[32] J. Le Feuvre, C. Concolato, and J. C. Dufourd. “Widgets Mobility”. In: *International Conference on Mobile Technology, Applications and Systems*. Nice, France, Sept. 2009.

[33] B. Pellan and C. Concolato. “Authoring of Scalable Multimedia Documents”. In: *Multimedia Tools and Applications* 43.3 (July 2009), pp. 225–252.

[34] C. Concolato. “Generation, Streaming and Presentation of Declarative EPG”. In: *EuroiTV*. Leuven, Belgium, June 2009.

[35] B. Pellan and C. Concolato. “Summarization of Scalable Multimedia Documents”. In: *Workshop on Image Analysis for Multimedia Interactive Services*. Londres, Angleterre, May 2009, pp. 304–307.

[36] C. Concolato, J. Le Feuvre, and K. Park. “An MPEG-based Widget System for CE and mobile devices”. In: *ICCE*. Las Vegas, Etats-Unis, Jan. 2009, pp. 1–2.

[37] B. Pellan and C. Concolato. “Adaptation of Scalable Multimedia Documents”. In: *ACM Symposium on Document Engineering*. São Paulo, Brazil, Sept. 2008, pp. 32–41.

[38] C. Concolato and J. Le Feuvre. “Playback of Mixed Multimedia Document”. In: *ACM Symposium on Document Engineering*. São Paulo, Brazil, Sept. 2008, pp. 219–220.

[39] B. Pellan and C. Concolato. “Scalable Multimedia Documents for Digital Radio”. In: *ACM Symposium on Document Engineering*. São Paulo, Brazil, Sept. 2008, pp. 221–222.

[40] B. Reiterer, C. Concolato, J. Lachner, J. Le Feuvre, J. C. Moissinac, S. Lenzi, S. Chessa, E. Fernández Ferrá, J. González Menaya, and H. Hellwagner. “User-centric Universal Multimedia Access in Home Networks”. In: *The Visual Computer* 24.7-9 (July 2008), pp. 837–845.

[41] B. Pellan and C. Concolato. “Spatial Scene Adaptation in Broadcast Environment”. In: *IEEE International Conference on Multimedia and Expo*. Hannovre, Allemagne, June 2008, pp. 389–392.

[42] B. Reiterer, C. Concolato, J. Lachner, J. Le Feuvre, J. C. Moissinac, S. Lenzi, S. Chessa, E. Fernández Ferrá, J. González Menaya, and H. Hellwagner. "User-centric Universal Multimedia Access in Home Networks". In: *Computer Graphics International.* Istanbul, Turquie, June 2008.

[43] C. Concolato, J. Le Feuvre, and J. C. Moissinac. "Design of an Efficient Scalable Vector Graphics Player for Constrained Devices". In: *IEEE Transactions on Consumer Electronics* 54.2 (May 2008), pp. 895–903.

[44] J. Le Feuvre, C. Concolato, and J. C. Moissinac. "GPAC, Open Source Multimedia Framework". In: *ACM Multimedia.* Augsburg, Allemagne: ACM New York, NY, USA, Sept. 2007, pp. 1009 –1012.

[45] C. Concolato, J. Le Feuvre, and J. C. Moissinac. "Timed-fragmentation of SVG documents to control the playback memory usage". In: *ACM Symposium on Document Engineering.* Winnipeg, Canada, Aug. 2007, pp. 121–124.

[46] B. Pellan and C. Concolato. "Media-Driven Dynamic Scene Adaptation". In: *8th International Workshop on Image Analysis for Multimedia Interactive Services.* Thira, Santorini, Greece, June 2007, pp. 67–70.

[47] M. Ransburg, H. Hellwagner, B. Pellan, C. Concolato, R. Cazoulat, S. De Zutter, C. Poppe, R. Van De Walle, and A. Hutter. "Dynamic and Distributed Adaptation of Scalable Multimedia Content in a Context-Aware Environment". In: *European Symposium on Mobile Media Delivery.* Alghero, Sardinia, Italy, Sept. 2006.

[48] C. Concolato, J. C. Dufourd, and J. C. Moissinac. "Representing 2D Cartoons using SVG". In: *SMIL Europe 2003.* Paris (France), Jan. 2003.

[49] Roger Pantos and William May. *HTTP Live Streaming.* Internet-Draft draft-pantos-http-live-streaming-19. Work in Progress. Internet Engineering Task Force, Apr. 2016. 54 pp. URL: `https://tools.ietf.org/html/draft-pantos-http-live-streaming-19`.

[50] Paul Asente, Mike Schuster, and Teri Pettit. "Dynamic Planar Map Illustration". In: *ACM Trans. Graph.* 26.3 (July 2007). ISSN: 0730-0301. DOI: `10.1145/1276377.1276415`. URL: `http://doi.acm.org/10.1145/1276377.1276415`.

[51] Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. “Diffusion Curves: A Vector Representation for Smooth-shaded Images”. In: *ACM Trans. Graph.* 27.3 (Aug. 2008), 92:1–92:8. ISSN: 0730-0301. DOI: `10.1145/1360612.1360691`. URL: `http://doi.acm.org/10.1145/1360612.1360691`.

[52] Pieter Simoens, Bojan Joveski, Ludovico Gardenghi, Iain James Marshall, Bert Vankeirsbilck, Miha Mitrea, Francoise Prêteux, Filip De Turck, and Bart Dhoedt. “Optimized mobile thin clients through a MPEG-4 BiFS semantic remote display framework”. en. In: *Multimedia Tools and Applications* 61.2 (Aug. 2011), pp. 447–470. ISSN: 1380-7501, 1573-7721. DOI: `10.1007/s11042-011-0849-3`. URL: `http://link.springer.com/article/10.1007/s11042-011-0849-3` (visited on 07/31/2016).

[53] Alberto Messina, Francisco Morán Burgos, Marius Preda, Skjalg Lepsoy, Miroslaw Bober, Davide Bertola, and Stavros Paschalakis. “Making Second Screen Sustainable in Media Production: The BRIDGET Approach”. In: *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*. TVX ’15. Brussels, Belgium: ACM, 2015, pp. 155–160. ISBN: 978-1-4503-3526-3. DOI: `10.1145/2745197.2755517`. URL: `http://doi.acm.org/10.1145/2745197.2755517`.

[54] Francisco Ganacim, Rodolfo S. Lima, Luiz Henrique de Figueiredo, and Diego Nehab. “Massively-parallel Vector Graphics”. In: *ACM Trans. Graph.* 33.6 (Nov. 2014), 229:1–229:14. ISSN: 0730-0301. DOI: `10.1145/2661229.2661274`. URL: `http://doi.acm.org/10.1145/2661229.2661274`.

[55] Pierpaolo Baccichet, Xiaoqing Zhu, and Bernd Girod. “Network-aware H. 264/AVC region-of-interest coding for a multi-camera wireless surveillance network”. In: *Picture Coding Symp.* 2006.

[56] Kiran Misra, Andrew Segall, Michael Horowitz, Shilin Xu, Arild Fuldseth, and Minhua Zhou. “An overview of tiles in HEVC”. In: *IEEE Journal of Selected Topics in Signal Processing* 7.6 (2013), pp. 969–977.

[57] M. Prangl, T. Szkaliczki, and H. Hellwagner. “A Framework for Utility-Based Multimedia Adaptation”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 17.6 (2007), pp. 719–728. ISSN: 1051-8215. DOI: `10.1109/TCSVT.2007.896650`.

---

[58] Weiping Li. "Overview of fine granularity scalability in MPEG-4 video standard". In: *IEEE Transactions on Circuits and Systems for Video Technology* 11.3 (2001), pp. 301–317. ISSN: 1051-8215. DOI: `10.1109/76.911157`.

[59] H. Schwarz, D. Marpe, and T. Wiegand. "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard". In: *IEEE Transactions on Circuits and Systems for Video Technology* 17.9 (2007), pp. 1103–1120. ISSN: 1051-8215. DOI: `10.1109/TCSVT.2007.905532`.

[60] P. Helle, H. Lakshman, M. Siekmann, J. Stegemann, T. Hinz, H. Schwarz, D. Marpe, and T. Wiegand. "A Scalable Video Coding Extension of HEVC". In: *Data Compression Conference (DCC), 2013*. 2013, pp. 201–210. DOI: `10.1109/DCC.2013.28`.

[61] Sung-Hee Park, Yeon-Bae Kim, and Yang-Scock Seo. "Multi-Layer Bit-Sliced Bit-Rate Scalable Audio Coding". In: *Audio Engineering Society Convention 103*. 1997. URL: `http://www.aes.org/e-lib/browse.cfm?elib=7259`.

[62] Thomas Stockhammer. "Dynamic Adaptive Streaming over HTTP –: Standards and Design Principles". In: *Proceedings of the Second Annual ACM Conference on Multimedia Systems*. MMSys '11. San Jose, CA, USA: ACM, 2011, pp. 133–144. ISBN: 978-1-4503-0518-1. DOI: `10.1145/1943552.1943572`. URL: `http://doi.acm.org/10.1145/1943552.1943572`.

[63] Pablo Cesar, Ishan Vaishnavi, Ralf Kernchen, Stefan Meissner, Cristian Hesselman, Matthieu Boussard, Antonietta Spedalieri, Dick C.A. Bulterman, and Bo Gao. "Multimedia Adaptation in Ubiquitous Environments: Benefits of Structured Multimedia Documents". In: *Proceedings of the Eighth ACM Symposium on Document Engineering*. DocEng '08. Sao Paulo, Brazil: ACM, 2008, pp. 275–284. ISBN: 978-1-60558-081-4. DOI: `10.1145/1410140.1410201`. URL: `http://doi.acm.org/10.1145/1410140.1410201`.

[64] A. Vetro. "MPEG-21 digital item adaptation: enabling universal multimedia access". In: *IEEE MultiMedia* 11.1 (2004), pp. 84–87. ISSN: 1070-986X. DOI: `10.1109/MMUL.2004.1261111`.

[65] S. Cho, G. Lee, B. Bae, K. Yang, C. H. Ahn, S. I. Lee, and C. Ahn. "System and Services of Terrestrial Digital Multimedia Broadcasting (T-DMB)". In: *IEEE Transactions on Broadcasting* 53.1 (2007), pp. 171–178. ISSN: 0018-9316. DOI: `10.1109/TBC.2007.891712`.

[66] Alex Zambelli. "IIS smooth streaming technical overview". In: *Microsoft Corporation* 3 (2009), p. 40.

[67] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K. Sitaraman. "BOLA: Near-Optimal Bitrate Adaptation for Online Videos". In: *CoRR* abs/1601.06748 (2016). URL: `http://arxiv.org/abs/1601.06748`.

[68] A. Beben, P. Wiśniewski, J. Mongay Batalla, and P. Krawiec. "ABMA+: Lightweight and Efficient Algorithm for HTTP Adaptive Streaming". In: *Proceedings of the 7th International Conference on Multimedia Systems*. MMSys '16. Klagenfurt, Austria: ACM, 2016, 2:1–2:11. ISBN: 978-1-4503-4297-1. DOI: `10.1145/2910017.2910596`. URL: `http://doi.acm.org/10.1145/2910017.2910596`.

[69] Federico Chiariotti, Stefano D'Aronco, Laura Toni, and Pascal Frossard. "Online Learning Adaptation Strategy for DASH Clients". In: *Proceedings of the 7th International Conference on Multimedia Systems*. MMSys '16. Klagenfurt, Austria: ACM, 2016, 8:1–8:12. ISBN: 978-1-4503-4297-1. DOI: `10.1145/2910017.2910603`. URL: `http://doi.acm.org/10.1145/2910017.2910603`.

[70] K. S. Yildirim, A. Ugur, and A. C. Kinaci. "Design and Implementation of a Software Presenting Information in DVB Subtitles in Various Forms". In: *IEEE Transactions on Consumer Electronics* 53.4 (2007), pp. 1656–1660. ISSN: 0098-3063. DOI: `10.1109/TCE.2007.4429266`.

[71] F. Gabin, M. Kampmann, T. Lohmar, and C. Priddle. "3GPP Mobile Multimedia Streaming Standards [Standards in a Nutshell]". In: *IEEE Signal Processing Magazine* 27.6 (2010), pp. 134–138. ISSN: 1053-5888. DOI: `10.1109/MSP.2010.938087`.

[72] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. "A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service". In: *Proceedings of the 2014 ACM Conference on SIGCOMM*. SIGCOMM '14. Chicago, Illinois, USA: ACM, 2014, pp. 187–198. ISBN: 978-1-4503-2836-4. DOI: `10.1145/2619239.2626296`. URL: `http://doi.acm.org/10.1145/2619239.2626296`.

[73] T. Lohmar, T. Einarsson, P. Fröjdh, F. Gabin, and M. Kampmann. "Dynamic adaptive HTTP streaming of live content". In: *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*. 2011, pp. 1–8. DOI: `10.1109/WoWMoM.2011.5986186`.

[74] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand. “Overview of the High Efficiency Video Coding (HEVC) Standard”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 22.12 (2012), pp. 1649–1668. ISSN: 1051-8215. DOI: `10.1109/TCSVT.2012.2221191`.

[75] Ngo Quang Minh Khiem, Guntur Ravindra, Axel Carlier, and Wei Tsang Ooi. “Supporting Zoomable Video Streams with Dynamic Region-of-interest Cropping”. In: *Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems*. MMSys ’10. Phoenix, Arizona, USA: ACM, 2010, pp. 259–270. ISBN: 978-1-60558-914-5. DOI: `10.1145/1730836.1730868`. URL: `http://doi.acm.org/10.1145/1730836.1730868`.

[76] Aditya Mavlankar, David Varodayan, and Bernd Girod. “Region-of-interest prediction for interactively streaming regions of high resolution video”. In: *Packet Video 2007*. IEEE. 2007, pp. 68–77.

[77] Vamsidhar Reddy Gaddam, Hoang Bao Ngo, Ragnar Langseth, Carsten Griwodz, Dag Johansen, and Pål Halvorsen. “Tiling of panorama video for interactive virtual cameras: Overheads and potential bandwidth requirement reduction”. In: *2015 Picture Coding Symposium, PCS 2015, Cairns, Australia, May 31 - June 3, 2015*. 2015, pp. 204–209. DOI: `10.1109/PCS.2015.7170076`. URL: `http://dx.doi.org/10.1109/PCS.2015.7170076`.

[78] Sheng Wei and Viswanathan Swaminathan. “Low Latency Live Video Streaming over HTTP 2.0”. In: *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*. NOSSDAV ’14. Singapore, Singapore: ACM, 2014, 37:37–37:42. ISBN: 978-1-4503-2706-0. DOI: `10.1145/2578260.2578277`. URL: `http://doi.acm.org/10.1145/2578260.2578277`.

[79] D. Li and L. Deng. “Multi-resolution Mechanism for SVG”. In: *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*. Vol. 2. 2009, pp. 139–143. DOI: `10.1109/APCIP.2009.171`.

[80] Ilya Grigorik. *High Performance Browser Networking: What every web developer should know about networking and web performance.* ” O’Reilly Media, Inc.”, 2013.

[81] Shanghong Zhao, Wei Tsang Ooi, Axel Carlier, Geraldine Morin, and Vincent Charvillat. “Bandwidth Adaptation for 3D Mesh Preview Streaming”. In: *ACM Trans. Multimedia Comput. Commun.*

Appl. 10.1s (Jan. 2014), 13:1–13:20. ISSN: 1551-6857. DOI: `10.1145/2537854`. URL: `http://doi.acm.org/10.1145/2537854`.

[82] Vamsidhar Reddy Gaddam, Ragnar Langseth, Sigurd Ljødal, Pierre Gurdjos, Vincent Charvillat, Carsten Griwodz, and Pål Halvorsen. "Interactive Zoom and Panning from Live Panoramic Video". In: *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*. NOSSDAV '14. Singapore, Singapore: ACM, 2014, 19:19–19:24. ISBN: 978-1-4503-2706-0. DOI: `10.1145/2578260.2578264`. URL: `http://doi.acm.org/10.1145/2578260.2578264`.

[83] Jack Jansen, Pablo Cesar, Rodrigo Laiola Guimaraes, and Dick C.A. Bulterman. "Just-in-time Personalized Video Presentations". In: *Proceedings of the 2012 ACM Symposium on Document Engineering*. DocEng '12. Paris, France: ACM, 2012, pp. 59–68. ISBN: 978-1-4503-1116-8. DOI: `10.1145/2361354.2361368`. URL: `http://doi.acm.org/10.1145/2361354.2361368`.

[84] D. Johansen, H. Johansen, P. Halvorsen, B. Olstad, C. Gurrin, and C. Griwodz. "Composing personalized video playouts using search". In: *Multimedia and Expo (ICME), 2010 IEEE International Conference on*. 2010, pp. 1534–1539. DOI: `10.1109/ICME.2010.5583236`.

[85] Jack Jansen and Dick C. A. Bulterman. "User-centric Video Delay Measurements". In: *Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. NOSSDAV '13. Oslo, Norway: ACM, 2013, pp. 37–42. ISBN: 978-1-4503-1892-1. DOI: `10.1145/2460782.2460789`. URL: `http://doi.acm.org/10.1145/2460782.2460789`.

[86] Rodrigo Laiola Guimarães, Dick Bulterman, Pablo Cesar, and Jack Jansen. "Synchronizing Web Documents with Style". In: *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*. WebMedia '14. João Pessoa, Brazil: ACM, 2014, pp. 151–158. ISBN: 978-1-4503-3230-9. DOI: `10.1145/2664551.2664555`. URL: `http://doi.acm.org/10.1145/2664551.2664555`.

[87] Dick C.A. Bulterman. "Interacting with Third-party Content: Is a Second Screen Enough?" In: *Proceedings of the 2014 ACM International Conference on Interactive Experiences for TV and Online Video*. TVX '14. Newcastle Upon Tyne, United Kingdom: ACM, 2014, pp. 1–2. ISBN: 978-1-4503-2838-8. DOI: `10.1145/2602299.2603000`. URL: `http://doi.acm.org/10.1145/2602299.2603000`.

[88] P. Houzé, E. Mory, G. Texier, and G. Simon. "Applicative-layer multipath for low-latency adaptive live streaming". In: *2016 IEEE International Conference on Communications (ICC)*. 2016, pp. 1–7. DOI: `10.1109/ICC.2016.7511550`.

[89] Wael Cherif, Youenn Fablet, Eric Nassor, Jonathan Taquet, and Yuki Fujimori. "DASH Fast Start Using HTTP/2". In: *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. NOSSDAV '15. Portland, Oregon: ACM, 2015, pp. 25–30. ISBN: 978-1-4503-3352-8. DOI: `10.1145/2736084.2736088`. URL: `http://doi.acm.org/10.1145/2736084.2736088`.

[90] Laura Toni, Ramon Aparicio-Pardo, Gwendal Simon, Alberto Blanc, and Pascal Frossard. "Optimal Set of Video Representations in Adaptive Streaming". In: *Proceedings of the 5th ACM Multimedia Systems Conference*. MMSys '14. Singapore, Singapore: ACM, 2014, pp. 271–282. ISBN: 978-1-4503-2705-3. DOI: `10.1145/2557642.2557652`. URL: `http://doi.acm.org/10.1145/2557642.2557652`.

[91] Christian Timmerer, Matteo Maiero, and Benjamin Rainer. "Which Adaptation Logic? An Objective and Subjective Performance Evaluation of HTTP-based Adaptive Media Streaming Systems". In: *CoRR* abs/1606.00341 (2016). URL: `http://arxiv.org/abs/1606.00341`.