A Pragmatically Designed Adaptive and Web-compliant Object-based Video Streaming Methodology

Implementation and Subjective Evaluation

Maarten Wijnants Gustavo Rovelo Peter Quax Wim Lamotte
Hasselt University – tUL – iMinds
Expertise Centre for Digital Media
Wetenschapspark 2, 3590 Diepenbeek, Belgium
firstname.lastname@uhasselt.be

ABSTRACT

The bulk of contemporary online video traffic is encoded in a traditional manner, hereby neglecting most, if not all, of the semantics of the underlying visual scene. One essential piece of semantic information in the context of video streaming is awareness of the objects that jointly constitute the scene. A canonical example of a benefit associated with such object awareness is the ability to subdivide a video fragment in respectively a background and one or more foreground objects. This paper reports on a pragmatically designed video streaming approach that exploits object-related knowledge in order to improve the real-time adaptability of video streaming sessions (manifested in the form of increased granularity in terms of streaming quality control). The proposed approach is completely compliant with present-day video codecs and HTTP Adaptive Streaming schemes, most notably H.264 and MPEG-DASH. Findings from subjecting the proposed video streaming technique to a comparative subjective evaluation suggest that scenarios exist where the presented approach holds the capacity to improve on traditional streaming in terms of user-perceived video quality.

Keywords

H.264; MPEG-DASH; MPEG-4; WebGL; object-based video

1. INTRODUCTION

Traditional video compression and streaming solutions are (largely) semantics-agnostic. For example, the majority of contemporary mainstream video codecs apply so-called *pixel-* or *frame-based* (de)compression. This implies that the encoder takes integral video frames as input and, via specific Rate-Distortion (RD) optimizations, divides the target encoding bitrate over the constituting frames and then over the constituting pixels of each frame. This happens without underlying knowledge of the individual objects that appear in the scene. The term "object" in this regard must be interpreted broadly, as it can refer to any entity that is distinguishable from the background (e.g., actors, salient props, and so on). Having access to object-related metadata paves the way for implementing a range of coding and streaming optimizations. One prime example that will be exclusively focused on in this paper is the additional versatility that it introduces in the streaming process. Being aware of the scene composition allows for deliberately distributing the bitrate budget over respectively the background and the foreground object(s). When starting our research, we hypothesized that, if executed properly, such an approach might benefit the perceptual quality of streamed video.

The MPEG Working Group recognized the potential coding benefits that are associated with object-based video compression as early as 1999, when they released their MPEG-4 specification with explicit support for the coding of video objects, both natural and synthetic [6, 13]. In particular, the MPEG-4 bitstream syntax encompasses so-called Video Object (VO) constructs and BIFS directives to represent (arbitrarily shaped) objects that are discernible in the video scene and to describe the scene composition, respectively. For each VO in a scene, the bitstream will carry separately encoded shape and texture information. Unfortunately, the object-based video part of the standard has witnessed very limited adoption in practice. We argue that this lack of adoption can be attributed to the disruptive nature of the proposed approach. The MPEG-4 VO standard breaks the traditional video processing workflow, at both encoding and decoding side, primarily due to the inclusion of the shape information in the encoded bitstream. As such, we believe that it suffered from poor hardware support and from the (costly) necessity to develop specialized software. Unsurprisingly, non-standardized object-based video coding solutions like the one proposed by Hakeem et al. [4] have seen even lower practical uptake.

In contrast, this article proposes an object-aware video streaming solution that is fully compliant not only with traditional video compression and streaming paradigms but also with prevailing Web standards. In particular, in this paper, our solution is shown to be compatible with H.264, arguably one of the most popular contemporary frame-based video codecs, and with MPEG-DASH (Dynamic Adaptive Streaming over HTTP), the de facto present-day streaming standard on the Web [7]. Although not yet confirmed experimentally, we are confident that the proposed approach could readily be incorporated in alternative compression and streaming setups as well. Seamless integration in existing workflows implies intrinsic inheritance of their advantages. As an example, the proposed solution fully pre-

MM '16, October 15–19, 2016, Amsterdam, The Netherlands.

 ISBN 978-1-4503-3603-1/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2964284.2964300

serves the server-side scalability, adaptive streaming and CDN compatibility benefits that are associated with MPEG-DASH.

Besides unambiguously establishing the technical and practical viability of the proposed object-based streaming approach in a Web browser context, this article will share, as an equally important research contribution, insights accumulated from a subjective video assessment test. This subjective evaluation was implemented as a comparative study, in which participants were asked to express their preference when presented with video sequences streamed using either the proposed object-based or a traditional approach (under identical bitrate limitations).

Although the proposed object-based streaming methodology is compatible (both theoretically and practically) with arbitrary video genres, we believe it to be especially well suited for those video scenarios in which one or more foreground objects substantially outweigh the scene background in terms of perceptual relevance to the viewer. Video conferencing, talk shows and interviews, and presentations (e.g., a news broadcast) are all prime examples of use cases belonging to the video streaming market niche that we will concentrate on in this manuscript.

In summary, the strengths of the proposed approach, either in relation to object-based MPEG-4 or in general, are the following: (i) the option to adaptively and independently stream the constituting entities of a video scene (e.g., background versus foreground objects), (ii) the portability benefits that are afforded by the Web-based implementation at client side, and (iii) the pragmatic nature of our solution that stems from its ability to exploit commodity video codecs (popularly implemented in both soft- and hardware) which, among other benefits, maximizes the range of supported playback devices.

2. CONCEPT AND METHODOLOGY

2.1 Object Segmentation

A fundamental prerequisite of any object-based streaming implementation is knowledge about the composition of the to-be-streamed video scene. To be able to distinguish between respectively background and foreground objects, for example, detailed information about the spatial location of these objects in the video clip over time is needed.

The problem of acquiring metadata pertaining to the location of in-scene video objects can be addressed in multiple ways. The results reported on in this article were obtained by manually segmenting the objects of interest from (the background of) the video scene as part of an offline pre-processing effort. Off-the-shelf products and software exist to facilitate this task. We found the rotoscoping tool of the Adobe After Effects software [1] to serve our object segmentation needs quite nicely. That being said, we are well aware that a more automated object segmentation scheme would be needed to render our approach economically viable and also to enable live streaming scenarios. Several unsupervised video segmentation algorithms are described in the academic literature that show promise in this regard (e.g., [12]).

The object segmentation task produces as output a number of distinct video signals, each depicting either a specific constituting video object or the residual background. All of these video fragments have the same spatiotemporal resolution as the original input (i.e., the unprocessed video). The video fragments for the objects hold the pixel data for that object over time, on a frame-by-frame basis, with the remainder of the pixels in each frame being set to a predefined color. For the video conveying the scene background, that same pixel color is used to fill up the holes caused by "cutting out" the foreground objects. All the content items that were prepared for the research described in this article use pure green (i.e., RGB (0, 255, 0)) as color value to represent pixels that have been segmented away. This is illustrated in Figure 1.

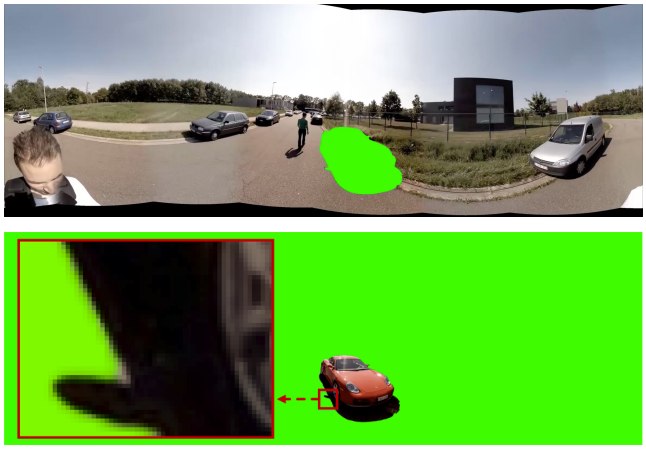

Figure 1: Segmented video scene involving a background and one single foreground object.

2.2 Encoding and Streaming Preparation

Each of the video fragments that emerge from the object segmentation operation is frame-based encoded in multiple qualities (see also Section 4.3). Then, conform to the general MPEG-DASH methodology, the resulting quality versions of both the isolated objects and the residual background are temporally divided in successive chunks and described by means of dedicated Media Presentation Description (MPD) manifests.

2.3 Distribution

The MPDs as well as MPEG-DASH video data that result from the encoding step are published simply by hosting them on an off-the-shelf Web server and are delivered to the client over HTTP in a pull-based manner. In line with the MPEG-DASH design philosophy, the client hereby retains total freedom to on-the-fly adapt, for example driven by prevailing network conditions, the quality in which the residual background as well as each of the isolated video objects is streamed during the media presentation. As the distribution task of the proposed methodology does not deviate from traditional MPEG-DASH streaming, neither conceptually nor technologically, it will not be elaborated on.

What is of interest in terms of content transport though, is the observation that the availability of dedicated video data for respectively the background and the constituting objects of a scene greatly extends the flexibility with which the streaming session can be implemented. In particular, it introduces an additional degree of freedom in terms of quality scalability by enabling the client to trade off the background for the foreground object(s) in terms of allocated streaming bitrate. This bitrate balancing effort could be settled depending on, for example, the relative contribution of the involved video streams to the user-perceived quality of the overall video scene.

2.4 Rendering and Playback

To be able to stream a video scene in the proposed methodology, the client first needs to fetch the MPDs that are associated with respectively the scene background and the foreground object(s). These manifest files inform the client about the different

qualities in which background and foreground object(s) are available, as well as about their associated bitrate requirements. From this point on, the client possesses all necessary information to enable the flexible implementation of the back- versus foreground streaming process discussed in Section 2.3.

Irrespective of the quality in which the constituent background and foreground object(s) are streamed, these video sources need to be combined somehow to recreate the original video scene. This *recompositing* operation is realized by resorting to the *chroma keying* paradigm. Intuitively speaking, this implies that pixels carrying the predefined (opaque) chroma value denoting an "empty" patch in the video fragment (see Section 2.1) are first made fully transparent, after which the different sources are stacked on top of each other in the appropriate order, on a frame-by-frame basis. When done correctly, the resulting video and the original, unprocessed video will depict exactly the same visual scene, yet the former will not necessarily exhibit the latter's largely uniform distribution of visual quality over respectively background and foreground object(s).

3. RELATED WORK

The reader is reminded that the work presented in this article does not aim to develop a new object-based video coding solution, but rather to re-use mainstream video coding approaches to realize, in a pragmatic and standards-compliant fashion, the proposed object-based streaming methodology. As such, we will not dwell on the low-level technicalities of object-based video codecs. Instead, the related work discussion will focus on the high-level mindset behind object-based video streaming, while also touching on the topic of object-based video playback control and on complementary video streaming optimizations.

3.1 Object-based Video Coding & Streaming

Wuenschmann et al. have compared the data rate requirements of H.264 frame-based versus MPEG-4 object-based coding of a scene consisting exclusively of rotating (synthetic) cubes [20]. It is concluded that object-based schemes hold the potential to surpass frame-based codecs with respect to coding efficiency, especially as the complexity of the to-be-coded video scene rises.

Vetro and Sun have proposed a method to model the RD characteristics of a lossy shape coding scheme [16]. Combined with existing texture-centric RD models, the proposed work enables joint rate control of respectively the texture and shape data of an MPEG-4 Video Object.

Vetro et al. have also discussed the additional degrees of freedom that object-based coding injects in (MPEG-4-powered) video transcoding frameworks [17]. The presented results highlight the increased adaptation flexibility that is unlocked by relying on object-based video compression (e.g., which objects need to be included in the transcoded result, and at what quality). Furthermore, potential RD gains by reducing the temporal resolution of individual Video Objects are demonstrated. However, it is also shown that combining objects with varying temporal resolutions in a single video scene might introduce composition issues.

A high-level discussion of the benefits that are associated with the object prioritization options inherent to an adaptive object-based streaming system is given by Goor and Murphy [3]. In particular, a system is envisioned that apportions resources (e.g., network bandwidth) among the constituent objects of a video scene proportionally to their relative priority or perceptual significance. Unfortunately, a concrete implementation of the weighted resource distribution concept is lacking, as are experimental or perceptual results.

Finally, an interesting application of the object-based streaming concept to the domain of Intelligent Transportation Systems is given by Hsiao et al. [5]. The described distributed system at capture side isolates vehicles in car traffic footage, compresses the resulting assets as well as the background using MPEG-4 object-based coding (hereby favoring the foreground objects in terms of assigned bitrate), and then transmits the encoded bitstream to a remote traffic monitoring module. This module in turn recomposes the streamed scene to implement traffic event detection and analysis, hereby benefiting from the high(er) visual quality of the vehicles compared to the scene background.

3.2 Object-based Video Playback

A number of comparable approaches are described in the academic literature that focus on object-mediated video browsing. Notable examples of such approaches are *Trailblazing* [9] and the scheme proposed by Nguyen et al. [10]. In essence, these approaches enable viewers to control the playback of a video clip via direct manipulation of the objects that it embeds. The direct manipulation in this case takes the form of spatially dragging an object along its movement trajectory in the video scene. During such dragging operations, the playback leaps to exactly that temporal instant in the video timeline when the manipulated object occupies the specified spatial position in the video scene.

While these approaches are concerned with video playback control rather than video streaming, there nonetheless exists a substantial conceptual match with the proposed methodology. In particular, they both fundamentally revolve around exploiting awareness of the composition of the video scene and, more precisely, of the objects that are featured in it.

3.3 Video Streaming Optimization

One specific way to look at the work presented in this article is in its capacity of a video streaming optimization. In this context, it is important to emphasize that the proposed methodology is applicable not only to traditional video formats (which is the focus in this paper) but also to panoramic or even 360 degree video. In fact, the frames depicted in Figure 1 are part of an equirectangularly projected 360 degree video capture. Given the extended spatial reach of panoramic footage, it could make sense to extract meaningful objects from the scene, disseminate those in full fidelity, and apply quality degradation to the residual background so as to reduce the streaming bandwidth requirements while still maintaining an acceptable video quality at receiver side.

An important research track in the area of panoramic video streaming is the *spatial tiling* concept (see, for example, [14]). The rationale of this concept consists of spatially segmenting the video footage into patches (typically rectangularly shaped) which can then be independently streamed. As such, it becomes feasible to assign variable bandwidth budgets to individual tiles depending on, for instance, the saliency of the video substance they contain [2]. The proposed methodology and the tiled streaming concept are complementary rather than mutually exclusive. Indeed, spatial tiling could be exploited to optimize the streaming of the background of an object-based scene [19]. Please also note that both approaches preserve compatibility with the adaptive streaming provisions offered by MPEG-DASH.

An alternative approach towards video streaming optimization is taken by the Scalable Video Coding (SVC) paradigm that enables the perceptual quality of a video sequence to be upgraded by adaptively streaming and applying so-called enhancement layers [15]. As is the case with spatial tiling, the proposed methodology and SVC are compatible, in the sense that the

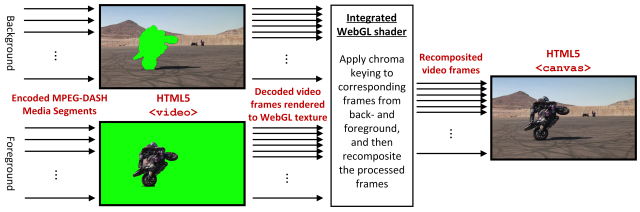

Figure 2: Chroma keying and scene recompositing.

latter could be applied to scalably distribute the background and/or constituting objects of an object-segmented scene.

4. IMPLEMENTATION

This section will focus exclusively on the implementation of the final part of the proposed methodology, namely the client-side rendering and playback of received footage in a Web browser context (see Section 2.4). The preceding steps in our end-to-end streaming pipeline are deliberately intended to reuse existing workflows and hence do not require specialized implementations.

4.1 Overview

The proposed object-based video streaming methodology is implemented as a fully standards-compliant Web application, with all of the client-side logic being handled in JavaScript. The MPDs describing the constituent entities of the to-be-streamed scene (i.e., background as well as isolated video objects) are fetched via AJAX, as is their video contents (packaged in the form of MPEG-DASH Media Segments). The client maintains a dedicated `HTMLVideoElement` per scene entity, which is fed with downloaded Media Segments via the W3C Media Source Extensions (MSE) specification [18]. Recall from Section 2.4 and Figure 1 that the resulting videos are not suitable for direct playback, as they still need to be composited together after first having been subjected to chroma keying processing. Abstractly speaking, the chroma keying processing task makes transparent those pixels in the decoded video frames that do not carry meaningful color data for the entity at hand, while the compositing task rebuilds the original video scene by superimposing the processed frames over each other in depth descending order. The outcome of the compositing task is rendered in an output `HTMLCanvasElement` for presentation to the user.

4.2 Chroma Keying

The chroma keying and scene compositing operations that form the cornerstone of our client-side methodology have been jointly implemented as an integrated WebGL shader (see Figure 2). In fact, two alternative fragment shader implementations have been developed to fulfill this compound task. Both expect to receive WebGL textures depicting temporally corresponding frames from the scene's composing entities as input and produce as output a single frame depicting the recomposited scene.

4.2.1 Classic Chroma Keying

The first WebGL fragment shader performs chroma keying in a classical manner. The shader iterates over all contributing entities in the scene (starting from the background and then continuing in depth descending order), extracts the color of the indexed pixel in the entity's video frame, applies the chroma keying operation on it, and then performs alpha blending with the previous composition state. The chroma keying operation itself first calculates the Euclidian distance between respectively the color of the indexed pixel and the color value that is used to represent transparency in the encoded video frame. The calculated Euclidian distance is then transformed into an alpha amount based on a lower and upper tolerance value (see Listing 1).

```
1  // vec4 pixClr = color of indexed pixel, int d = Euclidian distance,
2  // int x = dominant RGB channel of chroma value denoting transparency
3
4  if (d > tola) { d = 1.0; }
5  else if (tolb < d) {
6    // Reduce color intensity of chroma key value
7    pixClr[x] = pixClr[x] - min((1.0-d), (15.0/255.0));
8    // Map ]tolb,tola] interval to ]0,1] alpha value
9    d = (d-tolb)*(1.0/(tola-tolb));
10 } else { d = 0.0; }
11
12 pixClr[3] = d; // set pixel's alpha value
```

Listing 1: Classic chroma keying implementation.

4.2.2 Alpha Mask Optimization

The second chroma keying implementation combines the video object representation method described in Section 2.1 with an alpha mask as a complementary means to convey pixel opacity information. The alpha mask is baked into the video data that results from the object segmentation operation, in a top-bottom composition (see Figure 3). As such, video frames in the alpha mask approach exhibit twice the vertical resolution compared to those employed in the classic chroma keying implementation.

Implementation-wise, each decoded video frame (carrying at the same time image data and associated alpha mask) is rendered to a single texture, with separate texture coordinates being applied in the WebGL fragment shader to discriminate between the two conceptual data sources it encapsulates. The pixel color that is extracted from the lower part of the texture (any channel in the RGB color space will do as they will always hold equivalent values) is applied as the alpha channel of the corresponding pixel in the upper part of the texture. The scene compositing implementation is equivalent to the one described in Section 4.2.1.

4.2.3 Color Contamination Issues

The chroma keying module of our client-side implementation operates on video data that has undergone lossy (de)compression. This seemingly trivial observation can have substantial implications on the achievable perceptual accuracy of the (color-driven) chroma keying process. In particular, the lossy coding might cause imperfect preservation of the color information that is carried in the video. In the context of the proposed methodology, this color corruption issue manifests itself most relevantly in the form of potential "color bleeding" artifacts introduced at object segmentation edges. At such edges, there must ideally be a hard transition from the chroma keying color to (the color of) either the in-video object or the residual background. The lossy (de)compression however might cause this color boundary to fade (see the magnified area in the bottommost picture in Figure 1), which in turn complicates the chroma keying processing. In practice, the color contamination can cause segmented video objects to be surrounded with a uniformly colored contour in the recomposited scene (with the color of the contour corresponding with the applied chroma keying value). An illustration of this effect is given in the middlemost frame in Figure 5.

Both chroma keying implementations have been optimized to attenuate the perceptual impact of the color corruption issue. In the classic chroma keying solution, the WebGL shader reduces, for each semi-transparent pixel in the processed image, the magnitude of the dominant RGB channel of the chroma keying color (see Listing 1, line 7). While this introduces (modest) color deformation in the processed image, it also smoothens the colored

Figure 3: Video object representation plus alpha mask.

contour. On the other hand, in the alpha mask approach, the values read from the mask are discretely clamped to denote either full transparency or full opacity (as the lossy compression might cause black-versus-white boundaries in the alpha mask to be transformed into a grayish color). This optimization however cannot amend the color bleeding that occurs in the video object representation itself. A rather pragmatic solution to cope with the latter problem consists of combining an alpha mask that tightly fits the video object in the scene with video data in which the object boundaries have been shifted a few pixels outwards (see the magnified area in the object representation in Figure 3). This approach produces a spatial buffer area to absorb the color spill, with this buffer area subsequently being trimmed away by the closely fitting alpha mask.

4.3 Quality Adaptation Considerations

In theory, arbitrary types of quality adaptation can be applied to the video fragments that result from the object segmentation step in the proposed methodology. In effect, both spatial and temporal resolution modifications are theoretically supported, as is alteration of the Quantization Parameter (which intrinsically influences video output quality). In practice however, mixing video fragments that exhibit heterogeneous spatial or temporal fidelities holds the risk of introducing visual inconsistencies in the recomposited scene. In particular, spatial downsampling inherently hurts the accuracy of the carried color information, which in turn could cause alignment issues to arise. As an example, the surface area of a segmented object as encoded in respectively a downsized residual background and its full resolution video object representation might not match spatially. On the other hand, as has already been established by Vetro et al. [17], temporal resolution variation among video fragments contributing to a single scene might lead to the introduction of spatial patches in the recomposed frames for which no color data is available, this way effectively creating transparent "holes". Such a scenario arises, for example, when a high-framerate foreground object has changed its position in the scene, while its associated low-framerate residual background did not update yet.

4.4 Limitations Imposed By Web Browser

The client-side implementation is intended to be executed in a Web browser environment. There exist important hiatuses in the multimedia support offered by the targeted execution context, which considerably complicated our implementation and limits its achievable level of perceptual performance. First of all, the media APIs exposed by contemporary Web browsers prevent frame-accurate video seeking. This restriction forced us to read the constituting frames of a video while its playback is ongoing, via a timer-based approach (with the timeout period set according to the frame rate of the involved video). As the precision of JavaScript timers is known to be rather coarse, it is not guaranteed that every single input frame actually finds its way to the chroma keying module (or, conversely, that the chroma keying module never operates on duplicate frames). Secondly, even though media synchronization provisions are stipulated in the HTML5 standard, to date none of the mainstream Web browsers implement them. Our client-side methodology requires the video playback of the contributing entities in an object-based scene to be in sync, as the chroma keying and compositing operations need to be supplied with temporally matching frames from all involved entities for the recomposited result to be visually correct. To circumvent this functional deficiency, we resorted to a manual synchronization implementation in JavaScript which entails starting the playback of the involved videos in immediate succession and re-syncing them as soon as their playback time starts to diverge too heavily. While this heuristic approach works fine most of the time, it does not warrant frame-precise inter-video synchronization. To the best of our knowledge, exact video sync is (for the time being) impossible to achieve in plain JavaScript. Please note that this observation also substantiates our design decision in the alpha mask implementation to combine the object representation with its associated alpha mask in an integrated video stream instead of packaging both separately.

5. EVALUATION SETUP

Both an objective and subjective evaluation has been conducted with the goal of perceptually comparing the two object-based video streaming implementations described in Section 4.2, not only mutually but also with respect to traditional video streaming. This section will describe the evaluation setup.

5.1 Content Sample

The evaluation featured three distinct video fragments. Two of these fragments (i.e., clips "captain" and "concert") were taken from the *IRCCyN IVC 1080i* video quality database [11], while the third video (called "NTIA poolhall") was fetched from the *Consumer Digital Video Library* (http://www.cdvl.org/). All involved videos have a Full HD spatial resolution, have a playback duration between 8 and 10 seconds, are free of shot transitions (e.g., scene cuts), and were collected in raw YUV422 format. If present, the video fragments' audio track was dropped.

Content-wise, the three clips are quite similar, in that they all depict scenes featuring a human actor in front of a rather trivial background. In particular, the captain clip shows a man looking through a spyglass in front of an artificial fountain, the concert clip portrays musical performer Jean-Michel Jarre walking around on a stage and talking in his microphone, while the poolhall scene depicts a talking person seated in a restaurant. The first two video fragments did not undergo any content-related editing, whereas the last clip was converted into a single shot video fragment by removing the trailing scene of the pool table. The captain and poolhall clips were recorded with a statically positioned and oriented camera, while in the concert fragment the movement of the performer was tracked by rotating a fixed camera. Representative frame excerpts from the content sample are shown in Figure 4. We hypothesized that


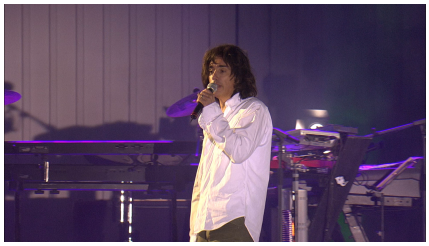


Figure 4: Representative frames taken from respectively the captain, concert and poolhall videos.

Table 1: Back- versus foreground bitrate, in kbps.

	Captain		Concert		Poolhall	
Low	200	400	200	500	350	750
Medium	300	700	400	850	450	1450
High	800	1200	800	1300	450	1950

the three considered videos could match well with the proposed object-based streaming approach, as we expected prospective viewers to focus their attention primarily on the foreground actor and much less on the (somewhat irrelevant) background (cf. the targeted video use cases mentioned in Section 1).

The spatial perceptual information (SI) and temporal perceptual information (TI) figures (as defined by Recommendation ITU-T P.910 [8]) of the content sample are as follows: the captain, concert and poolhall videos have a SI value of 33.15, 45.86 and 57.96, respectively, while their TI values respectively equal 20.35, 24.94 and 32.26. Intuitively speaking, these SI and TI measures quantify the intra-frame visual complexity and the amount of inter-frame motion, respectively. Whereas the items in our content set are conceptually and content-wise largely analogous, they nonetheless span a rather large area in the SI and TI continua (which range from 0 to infinity). We would have preferred to include videos with more comparable SI and TI values in our evaluation, yet this was found to be irreconcilable with the other requirements that we put forward for our content sample (i.e., the videos had to be available in uncompressed form and in addition had to depict conceptually largely similar content).

5.2 Content Preparation

For each item in the content sample, 3 discrete bitrate levels were determined (which will be abstractly denoted with the terms low, medium and high) so that each bitrate transition resulted in a clearly discernible quality difference as ascertained by an independent video quality expert. Then, via a manual and offline object segmentation step, the prominent actor in each of the three video clips was converted into a video object, with the remainder of the video scene being classified as belonging to the background. The three established bitrate amounts were finally distributed over respectively the scene background and the segmented foreground object for each of the processed videos (again in close consultation with the video quality expert). As can be read from Table 1, this was always done in such a way that the foreground object was allotted considerably more encoding bitrate compared to the background. The bitrate budget divisions as listed in Table 1 were enforced for both chroma keying implementations.

The practical considerations enumerated in Section 4.3 had two tangible implications on the content preparation. First of all, it was decided not to apply any temporal resolution scaling to the object-based video footage. This concretely implies that, for each of the three videos, the framerate of respectively the scene background and its associated foreground object always equaled the temporal resolution of the unprocessed video. Secondly, although compositing footage with varying spatial resolution might introduce object alignment issues, the consulted video expert nonetheless recommended to factor in spatial resolution manipulation in the evaluation. Therefore, whereas the foreground objects were always streamed in the original (i.e., Full HD) resolution, the scene backgrounds were spatially downgraded to a 1280x720 resolution (except for the concert background in the high bitrate setting, which the video expert preferred to stream in the original resolution, partly because the foreground object in this clip is spatially considerably smaller compared to the two other clips). To intrinsically mask any resulting perceptual errors, the scene background was in the evaluation always streamed integrally, without the foreground being "cut out" from it. This decision undoubtedly impaired the overall visual quality of the background (i.e., there is a bitrate cost associated with encoding the presence of the foreground object as opposed to a uniformly colored area), yet this penalty was deemed to be outweighed by the fact that it enabled the exploitation of spatial resolution as an additional degree of streaming freedom in the evaluation.

The content preparation actions described thus far yielded a total of 6 video clips: the unprocessed material (which immediately also served as background in the object-based footage) as well as the segmented foreground of three distinct videos. These were all VBR (Variable BitRate) encoded using the H.264 Main profile according to the target bitrates listed in Table 1. Please note that the bitrate sum of each back- and foreground pair in Table 1 served as target encoding bitrate for the corresponding non-object-based video. The resulting videos were next temporally divided into 2 second long MPEG-DASH Media Segments. Then, the object-based material was streamed via the proposed methodology and processed at client side using the two chroma keying implementations described in Section 4.2. The resulting output was screengrabbed (by periodically reading the raw pixel data of recomposited frames from the WebGL frame buffer) and subsequently losslessly encoded (Quantization Parameter value of 0) using the H.264 High profile. Although the non-object-based video footage could in theory have been evaluated directly, it nonetheless underwent the same processing (by treating it as an object-based scene consisting of only a single video object), just to exclude the impact of any perceptual effect introduced by the applied screen grabbing technique from the quality comparison results.

In summary, the content preparation finally amounted to 27 video clips (3 video fragments streamed in 3 different bitrates and in either a traditional or object-based fashion, with two chroma

Figure 5: Quality comparison: traditional (top) vs classic chroma keying (middle) vs alpha mask (bottom).

keying implementations being available for the latter streaming technique). Figure 5 displays a corresponding frame from the captain sequence when streamed using the three investigated techniques in the medium bitrate setting. Although complicated by the limited size of the images in print (please consult the paper's supplemental material for a veracious comparison), one should be able to discern that the detail and general quality of the background in the middle and bottom frame is lower compared to that in the traditional streaming snapshot. In contrast, the traditional approach is seen to perceptually suffer from a less sharp representation of the foreground actor, most notably his face. The color contamination issues described previously in Section 4.2.3 are noticeable in the middle image in Figure 5, especially around the person's hat in the right-hand side of the frame, whereas it does not appear in the object-based streaming implementation that applies the alpha mask optimization.

5.3 Subjective Assessment Methodology

The subjective part of our evaluation was implemented as a user study in which the Pair Comparison (PC) method as defined in Recommendation ITU-T P.910 [8] was applied. In the PC method, users are requested to mutually compare two video stimuli presented as a pair. The primary motivation for resorting to this evaluation technique is its high discriminatory power. As the user-perceptible visual differences that exist between the three tested streaming approaches were anticipated to be potentially small, Pair Comparison was deemed an excellent assessment method to quantify these differences. As a subordinate motivation, the PC mechanism involves a simple cognitive task that does not require specific knowledge or expertise from the assessor.

The subjective video quality assessment study logically encompassed three consecutive phases. In the first phase, some demographic information was collected by requesting the participant to fill in a short survey inquiring (primarily) about his or her video consumption habits. Then, the participant was handed written instructions about the test procedure. After verbally addressing any potential questions the participant had about the test procedure, he or she was then asked to complete a training session consisting of 5 representative test conditions (i.e., 5 video pairs exhibiting quality differences comparable to those appearing in the actual test later on). The objective of this trial run was three-fold: (i) familiarize the assessor with the test procedure in general and the employed rating scale in particular, (ii) counter potential learning effects, and (iii) stabilize the observer's opinion with respect to the range of quality differences that items in a pair might exhibit. The video content that featured in the practice session was not re-used in the actual test and the outcome of the session was discarded.

Once the training session was completed, the researcher left the room and the second phase of the user study commenced, which involved the participant conducting the actual experiment. Here, the participant was asked to assess a total of 57 pairs. For each of the three videos and predefined bitrate levels, the three investigated streaming techniques were paired in both the possible orders, this way giving rise to 54 pairs. Three control conditions, each consisting of two (arbitrarily chosen) identical video fragments, were randomly interspersed among these 54 meaningful pairs. The constituting items of a pair were always presented sequentially on the screen, with a fixed 2 second time interval being enforced between the two (during which the screen turned gray). After the presentation of the second item in each pair, the assessor was asked to grade the statement "I prefer the second video over the first" on a 5-point Likert scale ranging from "strongly disagree" to "strongly agree", with the middlemost value representing a neutral opinion. After the assessment of the 19th as well as the 38th pair, the participant was given the opportunity to relax (for as long as he or she saw fit) before continuing with the experiment. The pair presentation order was counterbalanced across test subjects (generalized Latin square experimental design) to minimize the impact of fatigue and other confounding factors on the aggregated set of observations.

While the participant was performing the actual PC test, the researcher observed his or her progress in real-time via an IP camera feed. As soon as the final pair had been rated, the researcher rejoined the participant in the study room to implement the third and final phase in the study consisting of a structured post-experiment interview. During the interview, the three tested streaming techniques were explained to the participant and then discussed. In its totality, a test session on average lasted approximately 1 hour, distributed nearly linearly over the three consecutive phases.

5.3.1 Apparatus and Setup

The user study was carried out in a dedicated room in our research institute. The apparatus of the study consisted of a

desktop PC running Windows 8.1, a 22" Full HD Samsung SyncMaster S22B300 monitor, and the Tobii EyeX optical gaze tracker (http://www.tobii.com/xperience/). The Pair Comparison test was implemented as a Web application. The desktop PC ran an Apache HTTP server to locally host this Web application as well as the 27 prepared video stimuli (see Section 5.2). The test application was executed in a commodity Web browser (i.c., Google Chrome version 48.0.2564.116 m) that was set to full screen mode. The brightness and contrast settings of the employed monitor were calibrated using the Windows "Display Color Calibration" tool.

5.3.2 *Participants*

A total of 18 users (4 female) participated in the subjective evaluation. All but two participants were between 20 and 30 years old, with the two outliers being older. All participants were either colleagues or university students. The subjects were screened for (corrected-to-)normal visual acuity and for absence of color vision deficiencies. The former test was implemented with a Snellen eye chart, the latter by resorting to Ishihara colored plates. On average, participants indicated to watch about 11 hours of video per week, with a PC being the most frequently used video consumption device, followed at considerable distance by smartphones and TV sets; tablets were found not to be a popular video consumption context in our population.

6. RESULTS

6.1 Objective Video Quality Assessment

The perceptual fidelity of the 27 content configurations (see Section 5.2) was objectively assessed by comparing their quality against that of their respective source signal (i.e., the unprocessed video on which the object segmentation was applied) using the PSNR and Structural Similarity (SSIM) metrics. The results are listed in Table 2. The PSNR and SSIM value spaces respectively equal [0, $+\infty$[and [-1,1], with higher values denoting better video quality. Please note that the reported objective figures were generated using a specialized Web implementation that overcomes the sync problems described in Section 4.4 by operating on decomposed frame sequences instead of video input. Contrary to video input, image-based input does allow tightly synchronized playback of the constituting entities of an object-based scene in the Web browser. Such frame-accurate entity playback sync is a prerequisite to enable objective frame-by-frame comparison with the source signal.

Table 2 reveals that the employed objective metrics found the tested video streaming techniques to be roughly comparable in terms of produced visual quality for the considered content sample. Therefore, the objective results will not be elaborated on.

6.2 Statistical Analysis of Subjective Data

A statistical analysis of the subjective results was conducted to ascertain whether the factors content (i.e., video fragment), bitrate and presentation order (i.e., when comparing streaming techniques A and B, did the participant first see them in either the AB or BA order) had a significant effect on the variable under investigation, in casu participants' preference with regard to the three compared streaming techniques. In the remainder of this subsection, we will denote the involved streaming schemes using the terms *traditional* (TR), (classic) *chroma keying* (CK) and *alpha mask* (AM).

Please recall that the evaluation adhered to a within-subjects experimental design (every participant assessed each of the 57 pairs) in which the presentation order of the factor combinations was randomized. After removing the results pertaining to the three control conditions (see Section 5.3) from the data set, the remainder of the recorded preference ratings was divided into three disjoint collections depending on the two video streaming techniques they applied to. The contents of the resulting groups was then halved by dropping the ratings that users expressed when comparing the streaming techniques in inverse presentation order (for any given video content and bitrate). In effect, the data resulting from reversing the presentation order was solely used as a reliability metric in the statistical analysis (see Section 6.3). Finally, for the three result groups separately, either the repeated measures ANOVA method or the Kruskal-Wallis non-parametric test was applied depending on whether Bartlett's test revealed a violation of the homogeneity of variances. If the ANOVA revealed a factor to have a statistically significant effect on the evaluation variable, a pairwise t-Test with Bonferroni corrections was applied as a post hoc method.

Concerning the alpha mask versus traditional streaming comparison results, the ANOVA did not show a significant main effect of any of the factors on users' preference for either of the two involved streaming techniques. For the comparison of the chroma keying versus the traditional approach, the ANOVA revealed a significant main effect of the bitrate factor on streaming technique preference ($F(2,153) = 3.75$, $p = 2.56e^{-2}$). However, the post hoc analysis did not show any significant differences among the three different bitrate levels. In the remaining result cluster (i.e., alpha mask versus chroma keying), the ANOVA detected a significant main effect of the content factor on streaming technique preference ($F(2,153) = 21.78$, $p = 6.15e^{-8}$). The post hoc analysis did show statistically significant differences between respectively the captain and poolhall videos ($p = 3.2e^{-8}$) and the concert and poolhall videos ($p = 9.9e^{-4}$). No statistically significant effect of the technique presentation order was discovered for any of the technique comparisons, nor were significant interactions among the three considered factors.

Figure 6 summarizes the results of participants' streaming technique preferences, accumulated across the three considered bitrate levels. In these plots, again only those ratings are retained that users expressed in the original (instead of in the inverse) streaming technique presentation order. The bar charts cardinally plot the number of times participants preferred one or the other paired streaming technique, without taking the amplitude of their preference into account. For each streaming technique combination and tested video clip, the number of neutral responses issued by assessors can be calculated by subtracting the number of votes jointly received by the two techniques from the aggregated number of comparisons, being 54. For example, for the alpha mask versus chroma keying comparison with the captain clip, a total of 28 neutral responses were registered. The numbers that are printed on top of the individual bars express the mean and standard deviation of users' comparison ratings when preference amplitude is taken into account (using numerical values 1 and 2 to respectively denote "(dis)agree" and "strongly (dis)agree" responses). Although we believe these results to be valuable, the reader is reminded that their statistical relevance was found to be limited.

6.3 Reliability of the Subjective Results

As a data reliability measure, the PC test included three hidden control conditions (in which identical videos were compared), while assessors also needed to grade all of the streaming technique comparison pairs twice, in alternative presentation orders.

Table 2: Objective video quality assessment results. Each table cell first lists the PSNR and then the SSIM value.

	Captain			Concert			Poolhall		
	TR	CK	AM	TR	CK	AM	TR	CK	AM
Low	31.223998	29.763731	29.651776	30.827201	27.220299	27.140646	37.110095	33.663130	33.704654
	0.856273	0.855087	0.851589	0.810569	0.780561	0.777704	0.946131	0.926638	0.927252
Medium	32.980461	31.224099	31.202164	33.431212	31.080416	31.050734	38.415972	34.142184	34.253600
	0.873295	0.865927	0.864808	0.837656	0.818014	0.816874	0.955235	0.933228	0.934946
High	35.116511	33.813269	33.843309	34.832957	32.246662	32.261694	38.876903	34.172041	34.295821
	0.896118	0.885625	0.885589	0.851363	0.829584	0.829355	0.958167	0.934060	0.935986

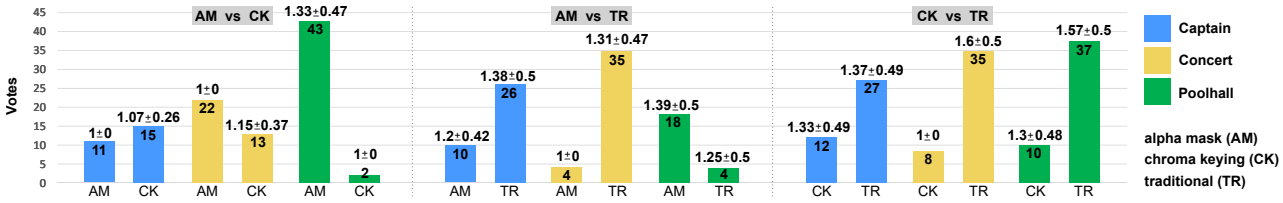

Figure 6: Absolute preference numbers for each of the streaming technique comparisons and tested video clips.

Analysis of the control condition data revealed that only once a strong preference for one of the compared videos was expressed by a test subject, whereas more than half of the grades (correctly) corresponded with a neutral opinion (i.e., 29 out of the total 54). We numerically encoded users' preference ratings for the control conditions as follows: 0 for neutral responses, 1 for (dis)agree preferences, and 2 for the strongly (dis)agree options. Using this coding scheme, the average preference value turned out to be 0.48±0.53. Please remark that in an ideal scenario, both values would be 0 (denoting neutrality). Concerning the results pertaining to the presentation order permuting, the analysis revealed little to no intra-subject variation. In absolute figures, in 204 out of the total 486 cases, participants expressed perfectly consistent preference ratings when comparing the same two streaming techniques in the two alternative presentation orders. In 121 of the residual 282 cases, participants maintained their preference yet with a different magnitude, whereas the remaining 161 cases yielded inconsistent ratings. By discretely mapping the numerical interval [-2,2] to the 5-point preference scale, the average rating difference for the 486 cases was found to equal -0.16±1.19. Again, the closer these values are to 0, the better.

6.4 Discussion

The statistical analysis of the subjective assessment findings did not turn out to be especially favorable to the proposed object-based streaming technique. However, we believe that this result needs to be put into proper perspective by also considering the outcome of the post-experiment interviews and participants' recorded gaze information.

As part of the post-experiment interview phase, we informed about the criteria participants applied when comparing the investigated video streaming techniques. It turned out that, even though we explicitly asked assessors to express their *preference* per paired streaming techniques (without explicitly mentioning the term *video quality* in this context), participants nearly unanimously interpreted their role in the experiment to primarily be that of a video quality assessor. In particular, many participants indicated that they actively searched for visual artifacts in the compared videos and, if present, were typically inclined to favor the video showing the least amount of artifacts. Recognized types of artifacts included general video blockiness or pixelation, the presence of contours surrounding segmented objects, and (overly noticeable) quality differences between back- and foreground. It is apparent that the application of such evaluation criteria hurt the appreciation of the tested object-based video sequences. Some participants explicitly mentioned that their appreciation of the object-based approach would likely be different if they would simply be watching videos at home at their leisure.

Participants' just described active scanning behavior (in search of visual artifacts) is objectively confirmed by the gaze tracking data. In particular, whereas many users initially tended to focus on the foreground objects, their attention was found to diverge to increasingly also include the scene backgrounds as the experiment progressed. This behavior of course unarguably defies the tested object-based streaming premise (i.e., shift bitrate allocation from back- to foreground, as this is the part of the video the viewer will most likely concentrate on). Besides artifact scanning, the observed gaze evolution can partly also be attributed to the large number of repetitions of the same 3 video clips (albeit in different bitrates or using different streaming techniques).

Many test users also indicated that, during the test, they had often tended to prefer the traditional streaming approach because it yielded the familiar scenario in which visual quality is rather uniformly distributed among back- and foreground. However, when we explained the rationale behind the investigated object-based approach, all 18 participants unanimously were found to be receptive to the idea. This is evidenced by the fact that nearly every test subject was able to independently devise at least one use case in which the proposed methodology could prosper. Some notable examples of imagined use cases were the streaming of a fashion show (with the models wearing the showcased clothes acting as foreground objects), video conferencing, task-centric or educational video applications (e.g., surgery training), and discourse scenarios (e.g., a human news presenter).

Finally, some participants remarked that the object-based streaming approach could have benefited from the inclusion of audio in the experiment. Especially for the concert video in the content sample, they thought that the presence of an accompanying audio track would have caused their attention to intrinsically be pulled more towards the foreground object.

6.5 Broader Implications

The subjective evaluation has revealed two broader video research implications. First of all, the contour artifact turned out to be either very annoying or very distracting (or both) for nearly all of the 18 test participants. The classic chroma keying

implementation considerably suffered from the presence of such artifacts, which we consider to be a determining factor for its slightly lower subjective appreciation compared to the alternative alpha mask implementation (although the difference was not found to be statistically relevant). We argue that the negative impact of the contour artifact on the viewing experience is likely to be extrapolatable to video streaming applications in general. Secondly, the quality differences as applied between respectively the back- and foreground in the evaluated object-based content sample were also classified by many test participants as being an undesirable video artifact. Quantifying the acceptable amount of quality variation among back- and foreground therefore represents an essential avenue for follow-up research.

7. CONCLUSIONS & FUTURE RESEARCH

This article has proposed and subjectively evaluated an object-based video streaming methodology that maintains full compatibility with contemporary frame-based video coding and HTTP Adaptive Streaming workflows. Statistically speaking, no decisive preference difference has been found to exist between respectively the proposed methodology and classical video streaming for the tested content sample. However, we believe that the reported subjective results correspond with the worst-case scenario (from our methodology's perspective), given the high amount of content repetition in the evaluation (causing assessors' attention to diverge to the background of the tested scenes), the lack of sound output, and the finding that participants overreacted to the presence of "artifacts" in the compared videos (although we really asked them to intuitively express their preference). This observation, combined with structurally elicited test participant feedback, causes us to conclude that the proposed methodology nonetheless holds the potential to outperform traditional streaming with respect to perceptual appreciation, at least in the targeted specialized video use cases involving salient foreground objects and plain backgrounds.

The work presented in this article has only scratched the tip of the iceberg with regard to pragmatic object-based video streaming. First of all, the reported experimental results exclusively concern simple video scenes consisting of a background and a single object of interest. Although it has already been empirically established that the proposed methodology is able to cope with the presence of multiple video objects in a single scene, such scenarios introduce complications which require further investigation. Secondly, given the detrimental effect of visual artifacts like object contours on user-perceived quality, integrating inpainting algorithms or other error concealment techniques in the proposed methodology seems advocated. Third, although not reported earlier in the article due to space constraints, the proposed methodology suffers from suboptimal performance on handheld devices like tablets. We therefore plan to conduct a workload analysis to amend computational bottlenecks. The final and probably most ambitious future research topic involves the design of an automated solution (e.g., based on emperically established heuristics) to optimally distribute the available streaming bandwidth over the constituting entities in an object-based video scene.

8. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement nr 610370, ICoSOLE ("Immersive Coverage of Spatially Outspread Live Events", http://www.icosole.eu).

9. REFERENCES

[1] Adobe After Effects CC tutorials. Use the Roto Brush. Online, https://helpx.adobe.com/after-effects/how-to/aftereffects-roto-brush-cc.html, 2016.

[2] P. R. Alface, J.-F. Macq, F. Lee, and W. Bailer. Adaptive coding of high-resolution panoramic video using visual saliency. In *Proceedings of the 1st International Workshop on Interactive Content Consumption at EuroITV 2013*, June 2013.

[3] S. A. Goor and L. Murphy. An adaptive MPEG-4 streaming system based on object prioritisation. In *Proceedings of Irish Signals and Systems Conference*, July 2003.

[4] A. Hakeem, K. Shafique, and M. Shah. An object-based video coding framework for video sequences obtained from static cameras. In *Proceedings of ACM MULTIMEDIA '05*, pages 608–617, New York, NY, USA, 2005. ACM.

[5] M.-H. Hsiao, H.-P. Kuo, H.-C. Wu, Y.-K. Chen, and S.-Y. Lee. Object-based video streaming technique with application to intelligent transportation systems. In *IEEE International Conference on Networking, Sensing and Control*, volume 1, pages 315–320, March 2004.

[6] ISO/IEC 14496-2:1999. Information technology – Coding of audio-visual objects – Part 2: Visual, 1999.

[7] ISO/IEC 23009-1. Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats, 2014.

[8] ITU-T P.910. Subjective video quality assessment methods for multimedia applications, April 2008.

[9] D. Kimber, T. Dunnigan, A. Girgensohn, F. Shipman, T. Turner, and T. Yang. Trailblazing: Video playback control by direct object manipulation. In *2007 IEEE International Conference on Multimedia and Expo*, pages 1015–1018, July 2007.

[10] C. Nguyen, Y. Niu, and F. Liu. Direct manipulation video navigation in 3D. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1169–1172, New York, NY, USA, 2013. ACM.

[11] S. Péchard, R. Pépion, and P. L. Callet. Suitable methodology in subjective video quality assessment: A resolution dependent paradigm. In *International Workshop on Image Media Quality and its Applications*, IMQA2008, Kyoto, Japan, September 2008.

[12] S. Poullot and S. Satoh. VabCut: A video extension of GrabCut for unsupervised video foreground object segmentation. In *2014 International Conference on Computer Vision Theory and Applications*, volume 2, pages 362–371, January 2014.

[13] A. Puri and A. Eleftheriadis. MPEG-4: An object-based multimedia coding standard supporting mobile applications. *Mobile Networks and Applications*, 3(1):5–32, June 1998.

[14] P. Quax, P. Issaris, W. Vanmontfort, and W. Lamotte. Evaluation of distribution of panoramic video sequences in the eXplorative Television project. In *22nd International Workshop on Network and Operating System Support for Digital Audio and Video*, NOSSDAV '12, pages 45–50, New York, NY, USA, 2012. ACM.

[15] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding extension of the H.264/AVC standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, September 2007.

[16] A. Vetro and H. Sun. An overview of MPEG-4 object-based encoding algorithms. In *International Conference on Information Technology: Coding and Computing*, pages 366–369, April 2001.

[17] A. Vetro, H. Sun, and Y. Wang. Object-based transcoding for adaptable video content delivery. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):387–401, March 2001.

[18] W3C Candidate Recommendation. Media Source Extensions. Online, https://www.w3.org/TR/media-source/, November 2015.

[19] H. Wang, V.-T. Nguyen, W. T. Ooi, and M. C. Chan. Mixing tile resolutions in tiled video: A perceptual quality assessment. In *24th International Workshop on Network and Operating System Support on Digital Audio and Video*, NOSSDAV '14, pages 25–30, New York, NY, USA, 2014. ACM.

[20] J. Wuenschmann, T. Roll, C. Feller, and A. Rothermel. Analysis and improvements to the object based video encoder MPEG 4 part 25. In *2011 IEEE International Conference on Consumer Electronics - Berlin*, pages 115–119, September 2011.