|  |  |
| --- | --- |
| **ITU – Telecommunications Standardization Sector**  STUDY GROUP 16 Question 6  **Video Coding Experts Group (VCEG)**  53rd Meeting: 20–26 February 2016, San Diego, US | Document VCEG-BA07 |

|  |  |  |  |
| --- | --- | --- | --- |
| Question: | Q.6/SG16 (VCEG) | | |
| Source: | **J. Ridge, M.M. Hannuksela (Nokia)** | Tel:  Email: | +16506180226  Justin.Ridge@nokia.com |
| Title: | **Virtual reality requirements for future video coding** | | |
| Purpose: | Information | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

This contribution considers the application of video coding to Virtual Reality (VR) content. It asserts that VR is gaining in use and popularity, and that future video coding standards should efficiently support VR. It further asserts that the differences between a “regular” video codec and one optimized for VR are not significant enough to justify a separate codec. The content of the contribution is essentially the same as MPEG input document m37709, and proposes specific use case and requirement text. It is understood that VCEG has not yet established a formal set of requirements for future video coding, however the authors anticipate that work may be conducted in a joint body such as JVET, and therefore an alignment of requirements between the parent bodies will be desirable.

## Introduction

Virtual reality (VR) is an application of video coding that is gaining in use and popularity to such an extent that future video coding standards should provide for VR requirements. VR places demands on the codec that make practical implementations and deployments challenging. This contribution lists requirements that, if satisfied, could improve user experience and simplify implementations. It is proposed to include the presented use cases and requirements into the Requirements for a Future Video Coding Standard.

## Background

### Virtual reality compared

Traditional uses of video coding involve the composition of a two-dimensional scene by the author. That could mean an average user pointing a camera, or the assembly of a scene by professional artists. The concept of 3D video generally means the addition of depth information to 2D video, thus the method of scene composition is not substantially changed.

More recently, the use of point-of-view cameras (e.g. GoPro) has become popular. The attractiveness of such cameras lies partially in the “vicarious” experience it provides to viewers, i.e. a sense of “being there” or “reliving” the situation. It is also partially due to the simplicity of using the system, including the capture process.

The limitation of point-of-view video is that the scene being viewed is limited to the one composed by the author – it is “see what I see”. It is likely that the viewer would look somewhere other than the author, either by choice or by reflex. This is particularly so when the author is making rapid movements of the head based on peripheral vision that the viewer does not see. Virtual reality intends to convert the “vicarious” experience into an “immersive” one, and permit the viewer to see what he or she wants to see, rather than the particular subset of the environment viewed by the author.

Free viewpoint TV (FTV) is a related but distinct topic. As with FTV, virtual reality is intended to transfer some of the scene composition ability from the author to the viewer. However, virtual reality tends to place the viewer at the center of the scene whereas FTV may not. Virtual reality also tends to be more interactive or responsive, e.g. rapidly responding to head or eye movements, rather than “watching” video from a given position for a longer period of time.

### Practicalities

Because composition of the scene is (at least partially) shifted from the author to the viewer, virtual reality video needs to be “over-captured”, i.e. more video is captured than can be viewed at any given time, to allow for (potentially rapid) changes in the viewer’s desired perspective. This could mean a limited extension to regular scene composition, i.e. “super-panorama”. At the other end of the spectrum, it could mean capturing 360 degrees to provide the viewer with complete flexibility, as is the case with Nokia’s OZO camera.

This provides the capture system with several technical challenges, including the sheer amount of video data and the atypical projections involved, e.g. fisheye lenses.

As with traditional video, it may be stored locally on physical media, or it may be streamed. In the streaming case, it is obviously desirable not to send the entire “over-captured” video bit stream, but rather the section the viewer is interested in. This presents particular challenges both to the codec, e.g. to enable identification of a particular perspective or “view” in the video bit stream, and to the delivery system.

It is not foreseen that these challenges would necessitate a significant deviation from “conventional” video codec design. This contribution asserts that it is preferable to incorporate VR requirements into the future video coding work, rather than create a distinct “VR-specific” video codec.

## Use cases

### Multicast streaming/broadcast

Virtual reality capture involves capturing the scene from many perspectives. Depending on the optics, this could involve a relatively small number of cameras, e.g. using fisheye lenses or similar, or it could involve a larger array of cameras covering a full 360 degrees, such as the eight lenses used in Nokia OZO.

The content from each camera could be “pre-stitched” so that any offset or overlap between cameras is taken into consideration and a single “scene” is composed prior to encoding. Alternatively, each “view” could be coded separately (or as a separate layer), potentially with some type of compensation between them.

Because there is no “feedback channel” in this use case, the capture device does not know which perspective the end user will take; thus capture must be complete (i.e. from every perspective). The quantity of video data being captured in this scenario means that some type of compression needs to be done near to the capture point, and that compression efficiency is extremely important.

From the client perspective, rendering the VR content may involve a spherical projection system, in which case the entire compressed bit stream is decoded and rendered. In this case, the decoder and/or renderer must take into account any warping or optical distortion from the capture process, so that the rendered video is displayed correctly.

Alternatively, rendering may involve displaying a subset of the captured video, e.g. using a VR headset. In this case, rather than rendering all the captured video, a particular view is extracted according to sensory feedback from the viewing apparatus. Such feedback occurs continually, so that as the viewer moves his or her eyes or turns his or her head, a different view is projected.

Because the capture and transmission is complete, delay is not a significant codec issue in this use case.

### Unicast streaming

The perspective requested by the viewer(s) is identified dynamically according to sensory feedback, and requested from the capture end. Thus a subset of the captured video, corresponding to the view that is desired by the viewer(s), is streamed to the decoding device(s). However, there is a tight delay constraint –whether or not the video is “live”, the rendered video must update promptly to avoid a “lag” between physical movement and the display.

### Single-user point-to-point

Live single-user applications mean that the view may be “negotiated”, i.e. the capture device potentially knows which view the user is interested in and can selectively request the far-end to capture and compress it. This use case is somewhat analogous to video conferencing due to the ability to negotiate, constraints on delay, etc. However, the system may still have to account for optical distortions and for the greater uncertainty due to the viewer’s ability to arbitrarily change viewing perspective.

## Future Video Coding requirements

1. The next-generation video codec shall efficiently code content captured using a variety of lenses and with a variety of projections.  
   *Example: Good compression efficiency using wide field-of-view and fisheye lenses, equirectangular panoramas, etc.*
2. The next-generation video codec shall code video efficiently when captured from a multiple lenses.  
   *Example: Exploit inter-lens redundancies to provide a compression efficiency advantage over parallel independent encodings.*
3. The next-generation video codec shall enable efficient extraction of a desired perspective from the compressed bit stream.  
   *Example: Extraction of perspective without decoding the entire bit stream.*
4. The next-generation video codec should enable efficient extraction of a desired perspective from the compressed bit stream without loss of coding efficiency.
5. The next-generation video codec should store parameters related to any correction or pre-processing due to optics, which may be useful to the rendering device in order to accurately reproduce the scene.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_