Deliverable

Project Acronym:	**ImmersiaTV**
Grant Agreement number:	688619
Project Title:	*Immersive Experiences around TV, an integrated toolset for the production and distribution of immersive and interactive content across devices.*

D3.4 Encoding and Decoding

Revision: 0.5

Authors:

Evgeniy Upenik, EPFL

Evangelos Alexiou, EPFL

David McNally, EPFL

Touradj Ebrahimi, EPFL

Delivery date: M22

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 688619

Dissemination Level

P	Public	x
C	Confidential, only for members of the consortium and the Commission Services	

Abstract: The video encoding and decoding components are responsible for the compression and subsequent decompression of the 360 omnidirectional video stream. This deliverable describes the status of the hardware and software encoding components delivered in Task 3.4.

REVISION HISTORY

Revision	Date	Author	Organisation	Description
0.1	31/08/2016	David McNally	EPFL	Document Edition
0.2	08/03/2017	David McNally	EPFL	Document Modifications
0.3	10/03/2017	Stephane Valente Maciej Glowiak	VS, PSNC	Review and final editing
0.4	28/08/2017	Evgeniy Upenik Evangelos Alexiou Touradj Ebrahimi	EPFL	Document Updates
0.5	13/10/2017	Maciej Glowiak	PSNC	Final corrections, styling

CONTRIBUTORS

First Name	Last Name	Company	e-Mail
Evgeniy	Upenik	EPFL	evgeniy.upenik@epfl.ch
Martin	Rerabek	EPFL	martin.rerabek@epfl.ch
David	McNally	EPFL	David.McNally@epfl.ch
Evangelos	Alexiou	EPFL	Evangelos.Alexiou@epfl.ch
Touradj	Ebrahimi	EPFL	touradj.ebrahimi@epfl.ch

CONTENTS

Revision History 1
Contributors 2
1. Introduction 4
2. Functionality 4
2.1. Codec Choices 4
2.2. Project Iteration One - Codec and Architecture 5
2.3. Project Iterations Two & Three - Outlook 5
2.4. Project Iteration Three – Outlook of Encoder 6
3. Code repository 8
4. Installation guide 8
5. Code documentation 8

1. INTRODUCTION

The video encoding and decoding (combined referred to as "codec") components are responsible for the compression and subsequent decompression of the 360 omnidirectional video stream. This video stream is constructed from multiple camera feeds by the stitching sub-system within the capture components of the ImmersiaTV distribution chain (See D3.2 – "Capture Components"). Compression is necessary in order to reduce bandwidth requirements while transmitting stitched 360 omnidirectional video content.

2. FUNCTIONALITY

The codec system must integrate at four different points within the ImmersiaTV process chain:

1. On the upstream side the encoder receives the output from the video stitching sub-system.
2. The encoded bitstream is then passed on to the content distribution network component of ImmersiaTV (CDN)
3. On the receiving end of the CDN, the decoder processes the encoded video bitstream
4. Outputs the decoded video frames to the multimedia server and home receptor for further processing, rendering and display.

In a codec system the encoder and decoder are to be considered as two integral parts of the same system. The ImmersiaTV architecture allows for ample freedom in terms of implementation and hardware characteristics of the platform hosting the encoder. In contrast to this, the decoder needs to be tightly integrated into the multimedia server as home receptor sub-system and is subject to a range of resource limitations and downstream compatibility constraints. As such, ImmersiaTV has adopted encoder technologies on the basis of the availability and compatibility of decoders which are suitable for integration within the adopted multimedia server as home receptor hardware and architecture.

2.1. Codec Choices

Based on the state-of-the-art in video codec technologies, two codec standards are considered suitable candidates for ImmersiaTV: H.264 (Advanced Video Codec or "AVC"), a mature and widely supported codec, and H.265 (High Efficiency Video Codec or "HEVC"), the newest standard in video coding technology released by MPEG[1]. While HEVC offers approximately two times better compression than AVC (for a given set quality target) and supports a range of desirable features not available in AVC, it is substantially more complex both on the encoder and decoder sides. This added complexity leads to integration challenges both on the encoder and decoder sides. In addition, the widespread adoption of HEVC (particularly on consumer end-user devices which, due to the high complexity of HEVC, require a hardware accelerated implementation of HEVC) has been dampened by HEVC's onerous licensing terms.

From the perspective of ImmersiaTV and given the constraints outlined above, it was decided to support AVC as the baseline codec technology both in project iterations one and two. Furthermore, and in view of the expected substantial quality benefits of an HEVC based video distribution solution, it was decided to investigate HEVC as a potential alternative in project iteration two and implement HEVC as the baseline codec solution during project iteration three.

[1] ISO/IEC JTC 1/SC 29/WG11

2.2. Project Iteration One - Codec and Architecture

Following the reasoning outlined above, ImmersiaTV project iteration one adopted AVC as the encoding and decoding solution. In iteration one, the content capture process breaks down into a series of temporally disconnected process steps: content capture, editing, stitching, encoding and delivery. In practice, ImmersiaTV integrated the encoding process side-by-side with the stitching process. This choice was taken because the stitching solutions provided by ImmersiaTV consortium members natively supported AVC. In view of optimizing the trade-off between encoded video quality and required bandwidth, the parameters controlling the AVC encoding process were tuned on the basis of both objective and subjective quality analyses.

2.3. Project Iterations Two & Three - Outlook

In view of an optimized video encoding process and in anticipation of migrating towards HEVC as the ImmersiaTV baseline codec technology, the encoding-delivery-decoding process chain has been refined as outlined in the following figure:

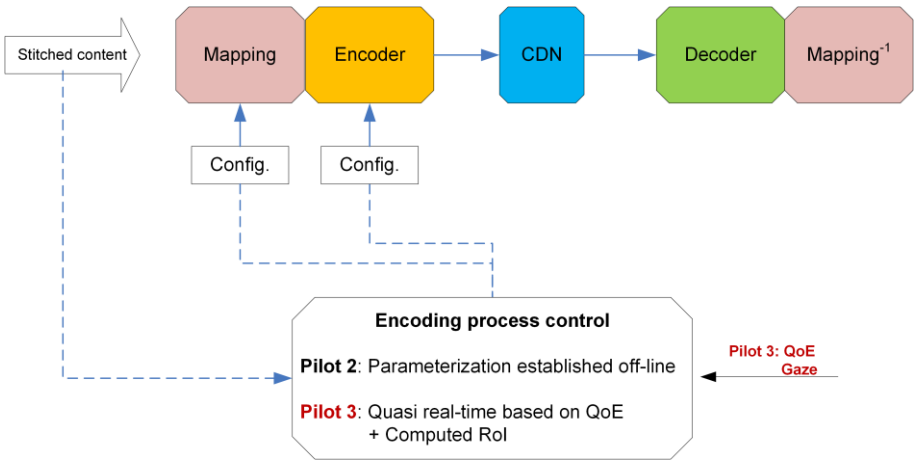

Fig. 1 – Refined encoder-deliver-decoder process chain

Project Iteration Two: In view of real-time and quality optimized encoding, particular care needs to be directed at how the stitched video content is presented to the encoder. That is, how the images in the stitched and uncompressed stream are mapped onto a rectangular frame for compression by the encoder. A suit of mapping algorithms have been implemented which are compatible with both the AVC and HEVC encoders freely available to ImmersiaTV.

For project iteration two and in order to reach real-time encoding, trade-offs need to be made within the parameter space defined by the following cornerstones:

1. Performance of the encoder computing platform (assuming that mapping and encoding are performed on the same physical machine – See Fig. 1)
2. Characteristics of the stitched content
3. Target quality, frame and bitrates
4. Constraints imposed by the chosen decoder (e.g. permissible decoding complexity and platform capabilities)

Achieving quasi real-time performance amounts to an investigation of the parameter space defined above and an appropriate optimization of the parameters controlling the mapping and encoding stages. The objective of this optimization is to achieve a coded video bit stream

delivering the best possible quality, in real-time and in compliance with the constraints imposed by the chosen target decoder(s).

2.4. Project Iteration Three – Outlook of Encoder

During this project iteration, the encoding process shall be dynamically controlled by feedback provided through a quality of experience analysis both at the encoder output and at the end user terminal. Iteration three will build on HEVC related work during iteration two in order to realizing this capability.

Several approaches are available to exploit particular properties of omnidirectional visual content for better compression.

Adaptive sliced delivery: Given with dynamic real-time information about the current gaze of the viewer, only a part of the whole picture, which covers the immediate field of view, is delivered to a receptor device. Architecture of such a system implies the use of a DASH-like server which divides an omnidirectional image/frame into slices and encodes each slice with several quality levels. The slices needed to cover the current viewport then delivered in the highest available quality, whilst the slices covering the rest of the omnidirectional frame are received in reduced quality. That is, when a viewer changes the direction of view he or she experiences a brief reduction of visual quality before the system adapts. However, adaptive sliced delivery depends on the current gaze of each end viewer and is not, therefore, suitable for broadcasting.

Visual Attention Coding: This approach lowers the entropy of the data representing an image by reducing spatial complexity of the regions which are of the least interest to the viewers. As it is depicted in Fig. 2, adaptive pre-filtering is applied after the optional re-mapping stage and before the base encoding block. The information about visual attention is received from QoE module and used to form a smoothing mask for a frame or a sequence of frames. Gaussian kernel can be used in the filter.

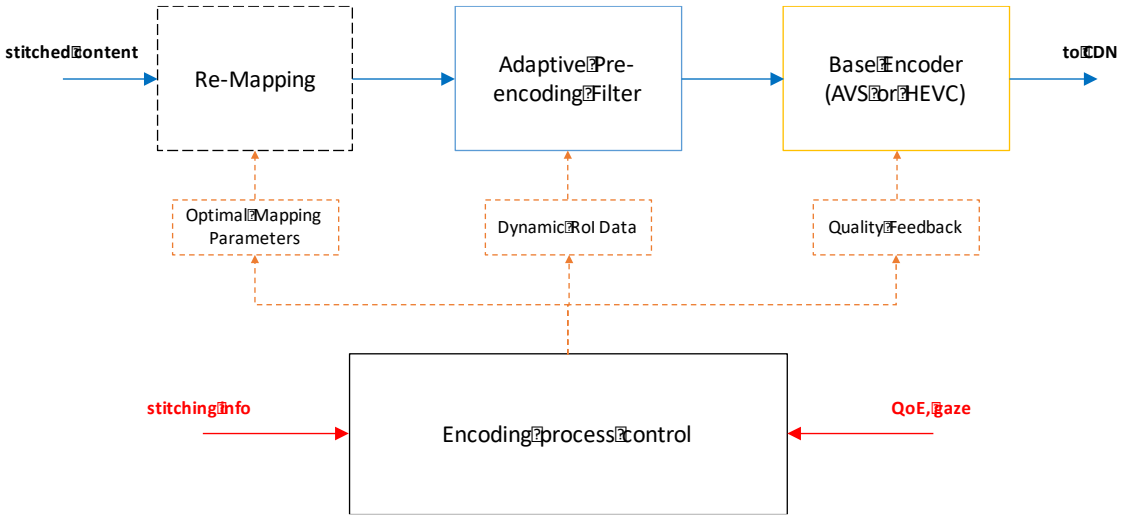

Fig. 2 – Encoder architecture diagram

Visual attention coding includes the following use cases:

Single user attention. In a unicast case, the information about current viewer gaze is transmitted back to encoder and used to form a filter which reduces the quality in of the regions not currently seen. The main different from the adaptive sliced delivery approach is that there is no need in a DASH-like server, and the system can be realised with simple pre-filtering.

Real-time combined attention. In a case of broadcasting, QoE module gathers the gaze information from all the viewers and after performing a statistical analysis provides a feedback to the encoding system with a visual attention map for each frame or a sequence of frames. The received visual attention map is then used to form a smoothing filter and to apply it to the video sequence.

Pre-obtained attention statistics. When one needs to broadcast pre-recorded omni-directional content, visual attention statistics can be obtained during the test viewing session or during the first broadcast if the recording is to be shown more than once. Similar to the previous use case, visual attention statistics is gathered by the QoE module. However, since there are no real-time restrictions, statistical analysis can be improved.

For the purposes of ImmersiaTV the second approach “Visual Attention Coding” appears to me more suitable and is recommended to be used.

3. CODE REPOSITORY

Remapping software:

In order to get access to Remap360 git-repository you need to perform the following steps:

1. Create an account at https://c4science.ch portal and inform us about your username. You will be then granted with the rights to the Remap360 repository.

2. Additional authentication is required to access git-repositories on c4science.ch. You need to configure your c4science account. In your profile settings, you can create a separate password to be able to access git via HTTP (VCS Password submenu). Or you can upload a public SSH key to access git via SSH (SSH Public Keys submenu).

3. The repository page is https://c4science.ch/diffusion/1930/.

Run the following command in you terminal:

```
git clone https://c4science.ch/diffusion/1930/remap.git
```

or

```
git clone ssh://git@c4science.ch/diffusion/1930/remap.git
```

4. Follow the instruction in README.md file.

4. INSTALLATION GUIDE

See code repository and associated documents

5. CODE DOCUMENTATION

See code repository and associated documents