INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2019/M47407
March 2019, Geneva, Switzerland

Source **Poznań University of Technology (PUT), Poznań, Poland**
Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea

Status **Input**

Title **Technical description of proposal for Call for Proposals on 3DoF+ Visual prepared by Poznań University of Technology (PUT) and Electronics and Telecommunications Research Institute (ETRI)**

Author Marek Domański*, Adrian Dziembowski*, Dawid Mieloch*, Olgierd Stankiewicz*, Jakub Stankowski*, Adam Grzelka*, Gwangsoon Lee**, Jun Young Jeong**, Jeongil Seo**,
* – Poznań University of Technology,
** – Electronics and Telecommunications Research Institute

1 Introduction

This document presents a technical description of compression technology prepared by Poznań University of Technology (PUT), Poland, and Electronics and Telecommunications Research Institute (ETRI), Korea, in response to Call for Proposals on 3DoF+ Visual [1].

The proposed HEVC-based technology employs rearrangement of the input multiview representation into coded representation which is composed of many views and layers.

For view synthesis, proprietary view synthesis software, developed by PUT and ETRI has been used, instead of RVS.

Table of contents

2 Overview of the proposed coding technology

The proposed coding technology is schematically depicted in Fig. 1.

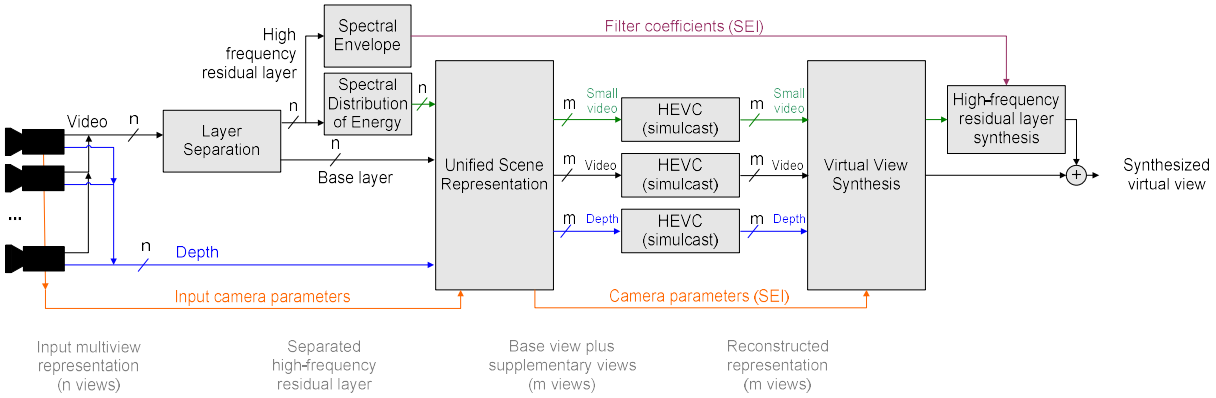

Fig. 1. The proposed representation and coding scheme.

The main idea of the proposed coding technology (Fig. 1) is to reduce the inter-view redundancy existing in the input multiview representation. For example, input representation may consist of multiple flat (rectangular) views with vastly overlapping fields of views. For another example, it may consist of a few 180-degree videos.

Therefore, for the sake of encoding, it is proposed to employ a different representation, called **Unified Scene Representation (USR).** The USR is attained by means of view synthesis which is performed to create a different, optimized set of views. For example, for multiple flat (rectangular) views with vastly overlapping fields of views, the USR is obtained by gathering the content of all views into a single omnidirectional view in equirectangular projection (ERP) format. Of course, a single omnidirectional video (with depth) is not enough to represent the whole 3-dimensional scene. Therefore it is allowed to send as many views (with depths) in USR as needed.

Moreover, it is allowed that some of the views in USR may convey usable information only in selected parts of the transmitted video. For example, this feature may be used to transmit only disocclusions in some of the views.

Moreover, the proposal exploits also a multi-layer approach, in which input video is spitted into layers in the spatial frequency domain. In the case of our proposal, the input video is split into two layers:

- **base layer**, which contains content that is spatially low-pass filtered, and that can be efficiently coded with classic predictive coding like HEVC.
- **residual layer**, which contains spatial high frequency residual content that can be represented jointly for a number of views.

Both layers are transmitted to the respective decoders and after decoding are summed together in order to produce reconstructed video. However, dedicated representation methods are used individually for the base layer and for the residual layer.

The content of the high-frequency residual layer is usually not compressed efficiently with classic predictive coding. Sample values of this layer are not correlated and resemble noise. Thus, the content of the residual layer is modeled as a random process which can be coded jointly among the views. The only parameters of this process are: spectral envelope and spatial distribution of energy.

The proposed technology is focused on the overall quality of experience of a 3-DoF+ system user. Therefore, we focused on the subjective quality of the synthesized virtual views presented to the viewer.

Proposed decoder/syntheser is not overcomplicated and allows to synthesize the virtual view without time-consuming postprocessing of the decoded views. Moreover, it provides scalability – by reducing the number of supplementary views the processing time and view synthesis complexity can be easily reduced.

3 Technical description of the implemented encoder

3.1 High-frequency residual layer separation

The separation of layers occurs at the very beginning of the processing as a result of motion-compensated temporal filtering [2]. Each frame of each view is processed independently. Block-based motion estimation is performed in order to find the motion vectors pointing to frames neighboring in time (3 previous and 3 next frames). The matched blocks are low-pass filtered. The process yields low-frequency base texture layer which is fed to the base encoder, whereas the remaining high frequency residual part of the input video is fed to the residual layer encoder. The layer separation process is entirely automatic.

The content of the high-frequency residual layer is usually not compressed efficiently with classic predictive coding. The content of the residual layer is modeled as a non-stationary random process which can be coded jointly among the views. As mentioned in the previous section, the only parameters of this process are: spectral envelope and spatial distribution of energy.

The spectral envelope is estimated from energy-normalized residual layer with the use of technique similar to LPC. The result is a set of IIR filter coefficients (in horizontal and vertical direction) which are coded with use of LAR (log-area-ratio) 16-bit representation. The proposed technology allows for coding of residual layer for all views or only for one selected view. In the latter case, residual layer in the missing views is synthesized.

Spatial energy distribution of the residual layer is estimated with use of block-based processing. The residual video is divided into rectangular non-overlapping blocks. In each of those blocks, energy is measured. Energy values, associated with respective blocks, constitute an image of spatial energy distribution, whose resolution is smaller than resolution of the input video. Spatial energy distribution is coded with use of HEVC-based coder.

3.2 Unified Scene Representation

3.2.1 Idea

It is proposed to represent the scene by one base view and a set of supplementary views. The supplementary views contain information unavailable (occluded) in the base view (Fig. 2). Supplementary views can be located in the same viewpoint as the base view (central supplementary views) or in different viewpoints (e.g. at the left and right side of the base view).

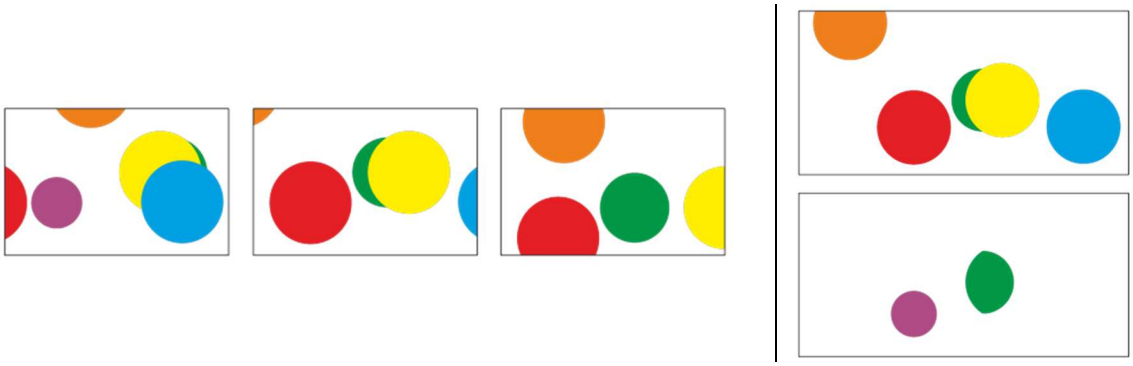

Fig. 2. Left: 3 input views, right: base view (top) and supplementary view (bottom).

3.2.2 Base and supplementary views synthesis

The base view and supplementary central views are created during one view synthesis process. Points from all the real views are projected into the base viewpoint. As opposed to the typical virtual view synthesis, where points representing occluded objects are omitted, in proposed method these points are stored in supplementary central views.

The base view is the typical virtual view, containing only the non-occluded information. The first supplementary central view contains points occluded in the base view. The second – points occluded by the objects from base view and first supplementary central view. In general, *i*-th supplementary central views contains information occluded in (*i*-1)-th supplementary central view.

The non-central supplementary views are created in three steps. In the first step, the supplementary view is synthesized using information from all the real views. Then, the base view is projected into the supplementary view. In the third step, for every point of the supplementary view it is checked, whether it was synthesized also from the base view. If so, that point is removed from the supplementary view.

3.2.2.1 General remarks

- Large areas of the supplementary views are empty, so inpainting of holes in the supplementary views should be disabled to avoid filling them.
- The angle of view of the base camera may be greater than for input cameras. Therefore, some areas close to the boundaries of the base view are empty (they were not seen in any of the input views). In order to not fill them using wrong information, extrapolating (by inpainting) near the base view boundaries is also disabled.
- Different real views may have different color characteristics (especially for natural multiview sequences), which have to be equalized before creating base and supplementary views. We propose to use a fast color correction technique. The global color difference between points projected from two real views is calculated as the mean ratio (averaged for the entire image) between color component projected from one view and color component projected from the second one [4]. The algorithm is performed separately for all color components, e.g. Y, C_B, C_R. In order to equalize colors of points projected from any real view *i*, color component values projected from view *i* are multiplied by the mean ratio between view *i* and reference view (the view acquired by the closest real camera to the virtual one).

3.2.3 Depth unification

For the sequences without inter-view consistency of the depth maps the additional step is required before extracting base and supplementary views. In this step the cross-view synthesis is performed in order to project depth values from all N into each of N input depth maps. After this step, for all the points in each depth map there is a list of depth values, projected from various input depth maps.

1. In order to provide the inter-view consistency, each point is processed in the same way:
2. All the depth values are sorted in descending order.
3. If *n* smallest depth values are similar (difference smaller than a threshold) go to step 6; else go to 3.
4. Remove first (smallest) depth value from the list.
5. If number of the elements in the list is smaller than n, go to step 5; else go to step 2.
6. Restore all the removed values to the list, decrement *n* and go to step 2.
7. The new depth value for analyzed point is an average value of these *n* values.

3.2.4 Location of base viewpoint

In order to ensure good visibility of most objects in the scene, it is recommended to locate the base viewpoint in the middle of the multicamera system.
For omnidirectional sequences, where real cameras are placed on a sphere or different volume, the base viewpoint should be placed in the center of that volume.
For non-omnidirectional sequences, the base viewpoint should be located in the middle between the leftmost and rightmost camera (along the horizontal axis) and in the middle between the topmost and bottommost camera (along the vertical axis). Along the z-axis, the base viewpoint should be located at least as far from the scene, as the farthest real camera.

3.2.5 Location of supplementary viewpoints

In proposed approach there are no constraints on number and arrangement of the supplementary views. However, performed tests demonstrated, that 4 supplementary views (left, right and two supplementary central views) are sufficient for realistic cases.

Both supplementary central views should be located in the same viewpoint as the base view. The left and right viewpoints should be located at least as far as the leftmost and rightmost real camera, respectively. Moreover, they should be shifted along the z-axis into the scene (at least as far as the closest real camera). For omnidirectional sequences, where no left/right dependencies are defined, the region of interest (e.g. people in TechnicolorMuseum sequence) should be arbitrarily chosen before supplementary viewpoints location. It is assumed, that the line connecting base viewpoint and chosen ROI is the z-axis, thus left and right viewpoints can be located.

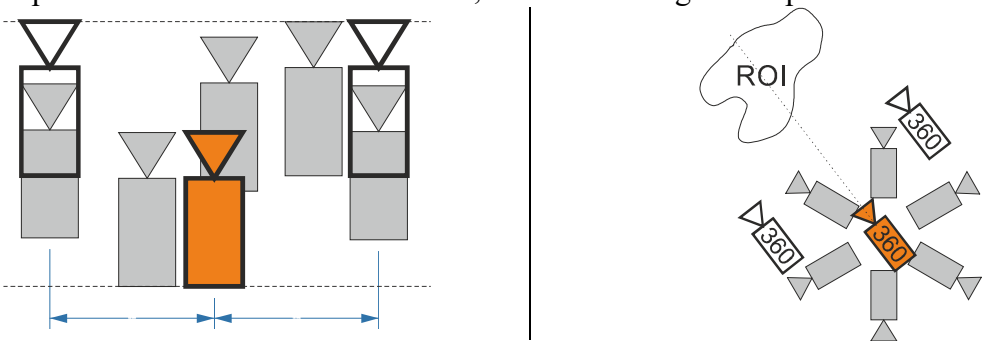

Fig. 3. Location of base viewpoint (orange) and supplementary left and right viewpoints (white) for non-omnidirectional (left) and omnidirectional sequence (right); input viewpoints are colored in grey.

3.2.6 Parameters of base and supplementary views

For omnidirectional sequences, base and supplementary views should be omnidirectional (360° × 180°). For other sequences they should contain information from the whole scene, thus the angle of view of the base and supplementary cameras should be increased. In this step perspective artifacts should be avoided, so the focal length of the virtual cameras which capture base and supplementary views is similar (e.g. the same) as the focal lengths of the real cameras. Therefore, angle of view increase is performed by increasing resolution of the view and shifting the principal point of the camera without any change of camera focal length.

3.2.7 Supplementary views for test sequences

Performed experiments showed, that optimal number of supplementary views is different for different sequences and it depends on scene difficulty (number of objects in the scene, their size and relative positioning).
TechnicolorMuseum sequence contains the most difficult scene (in term of virtual view synthesis), thus 3 supplementary views are required (left, right and 1 supplementary central view). For

IntelKermit only left and right supplementary views should be sent to the viewer. For TechnicolorHijack and TechnicolorPainter there is no need for sending left and right supplementary views and only 2 supplementary central views are required. For ClassroomVideo 1 supplementary central view is sufficient.

3.3 Preparation of views for encoding

3.3.1 Preparation of base view

Plain areas are easier to predict thus encode. Therefore, the edge between areas with texture projected from the input views and areas with no texture should be eliminated. Of course, for each point it is necessary to send information, whether the point was projected from any view or not. In proposal it is signaled by depth values, so the edge in base view's texture can be removed. In order to eliminate these edges, simple inpainting is performed (color of nearest non-zero pixel is copied to the interpolated one). That approach would result in many edges (along the inpainting direction) and temporal inconsistency, what would cause compression performance reduction. Therefore, we proposed to average interpolated color values both spatially and temporally. Spatial averaging is performed for 1-dimensional mask (e.g. 7×1 or 1×7 pixels), perpendicular to the direction of inpainting. Temporal averaging is performed by using weighted average of current color of each pixel and collocated pixel in the previous frame. Both average operations are performed only for interpolated areas.

3.3.2 Preparation of supplementary views

3.3.2.1 Cracks filling

During the virtual view synthesis the crack artifacts (1-pixel-wide holes) appear – both for base view and supplementary views. The cracks in the supplementary view are visible as small regions with no information or with significantly higher depth value than in their neighborhood. Crack artifacts in the base view may cause appearing of pixels with lower depth value than neighboring pixels in the supplementary view.

Both effects reduce the synthesis quality and encoding performance. Therefore, they have to be eliminated. In proposed approach the depth value of each pixel of the supplementary view is compared with depth value of its neighbors (both in vertical and horizontal direction). If both neighbors have higher depth value or both neighbors have lower depth value, the depth value for analyzed pixel and its color are updated by an average value of both neighbors.

3.3.2.2 Spatial hierarchical complementation of supplementary views

The supplementary view contains only the information occluded in the base view. Thus, most of its area is empty. However, there are many edges between empty and non-empty regions of supplementary view. If such edge is horizontal/vertical and it is located at the boundary of the CU, it may be encoded more efficient than if it has random shape and it is located inside the CU.
Therefore, in the second step we try to reduce the number of edges inside CUs by enlarging non-empty regions into the CU grid.

In the proposed algorithm, regions in supplementary views are enlarged using information (depth and texture) from collocated regions in the base view. However, information from the base view is copied only if the average depth value of analyzed region in the supplementary view is similar (smaller than a threshold) to the average depth value of collocated region in the base view. In this case all the empty pixels in analyzed block in the supplementary view are filled using depth and color from the base view.

The algorithm starts from block size 8×8 pixels and iteratively doubles block size up to 64×64.

3.3.2.3 Temporal hierarchical complementation of supplementary views

Arrangement of non-empty regions in the supplementary views changes over time. Therefore, in order to improve encoding performance the temporal complementation of supplementary views should be performed. In proposed approach, adaptation for the hierarchical GOP was assumed, where all the frames except for I-frame and one B-frame have to references: one previous and one forward frame.

The temporal complementation is performed only for empty pixels of the analyzed B-frame. If a pixel in analyzed B-frame is not empty (it has depth value and color), it is not modified. For every empty pixel in the B-frame it is checked, whether the collocated pixels in previous and forward frame are empty or not. If both are empty, analyzed pixel remains empty. If one of them is not empty, it also remains empty, because it was assumed, that encoder will use the other reference. However, if both collocated pixels are not empty, analyzed pixel is modified and its updated depth value and color are copied from the collocated pixel in the base view.

3.4 Implementation details

The virtual view syntheser was written in C++, VisualStudio 2017 for Windows platform. For decoding purposes, the libavcodec library was used.
In order to synthesize virtual views, _AVS_final.exe (in executables/synthesis directory on SSD) should be used. It requires .cfg file (included in .zip file for each rate and sequence). Our syntheser may operate on encoded input data, therefore the input bitstreams do not have to be decoded before synthesis. To execute synthesis, _AVS_final.exe xx.cfg command should be used. Encoded data are handled using libavcodec library, thus 4 required .dll files are included in executables/synthesis directory.

3.4.1 Configuration file for decoder / syntheser

```
NumberOfInputViews                  2
NumberOfOutputViews                 6

NumberOfFrames                      120
StartFrame                          0

RealCameraParameterFile             CV.txt
VirtualCameraParameterFile          camparams/CV_posetrace.txt

Width                               4112
Height                              2048

Format                              Omni    # Omni / Perspective
AOV_Left                            -180    # AOV - Angle of View
AOV_Right                           180
AOV_Top                             90
AOV_Bottom                          -90

ZNear                               0.8
ZFar                                1000
ViewChromaSubsampling               420
DepthChromaSubsampling              420
ViewBitsPerSample                   10
DepthBitsPerSample                  10

Compression                         HEVC    #HEVC or none

SynthesisMethod                     3       # 1 – pixel-based projection
                                            # 3 – triangle-based projection [3]
DepthBlendingThreshold              10
RemoveGhostEdges                    2       # elimination of artifacts caused by blurred edges
                                            # in the input views (0: disabled)
```

```
DepthBasedPrioritization            1       # if 1: overriding view priority for pixel if depth
                                            # difference is greater than DepthBlendingThreshold
BlurEdges                           1       # depth-based filtration of texture edges (0 / 1)

Input0 {
        CameraName              A_c00
        View                    P06_SA_R3_Tt_c00.bit
        Depth                   P06_SA_R3_Td_c00.bit
        AOV_Left                -180.7031
        AOV_Right               180.7031
        Priority                1               # 1 – highest priority, ...
}
Input1 {
        CameraName              A_c00
        View                    P06_SA_R3_Tt_c01.bit
        Depth                   P06_SA_R3_Td_c01.bit
        AOV_Left                -180.7031
        AOV_Right               180.7031
        Priority                2
}

Output0 {
        View                            P06_SA_R3_Tt_p02_2048x2048.yuv
        ViewBitsPerSample               8
        ResidualLayer                   CV_residual_2048x2048_10bps.yuv
        Width                           2048
        Height                          2048
        Format                          Perspective
        CameraName [
                A2_p0
                A2_p1
                …
                A2_p119
        ]
}
```

Remarks:

- SynthesisMethod was set to 3 for fully-synthetic sequences (ClassroomVideo and TechnicolorMuseum) and 1 for the rest,
- DepthBlendingThreshold was set to 40 for both perspective/natural sequences (TechnicolorPainter and IntelKermit), 20 for TechnicolorMuseum, 10 for ClassroomVideo and 4 for TechnicolorHijack,
- DepthBasedPrioritization was enabled for ClassroomVideo and TechnicolorHijack,
- Priority for base view was set to 1, supplemental views had lower priorities.

4 Synthesis of virtual view

4.1 Overview

In our proposal, we use a proprietary virtual view synthesis method that is different from RVS. The reason behind using different method is, first of all, the addition of omitting of regions that shall not be used in the final view synthesis. Moreover, other numerous improvements that increase the subjective quality of synthesized views were added, including the input views prioritization and the enhanced inpainting for omnidirectional views.

The proposed method is an extension of the triangle-based Multiview Synthesis method [3]. The overall scheme of the proposed method is depicted in Fig. 4. Each input view is projected to a virtual view separately. Then, data projected from all views are merged (regarding processing priorities) in order to produce the final virtual view. Disoccluded areas are filled from the further input views rather than inpainted.

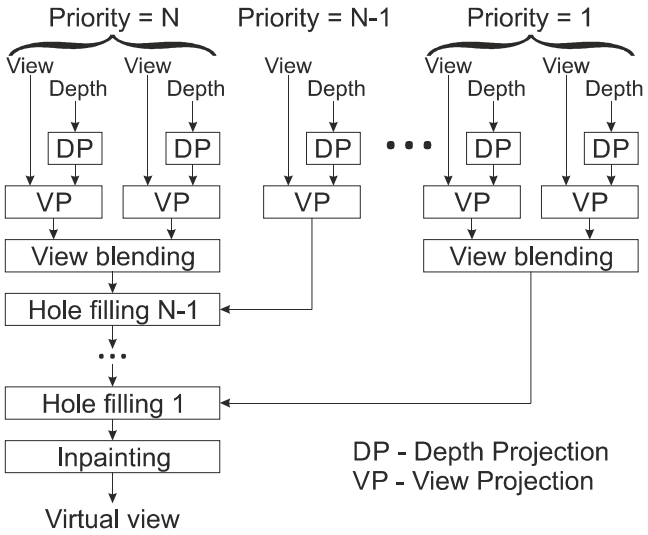

Fig. 4. The scheme of the proposed view synthesis method.

The input views with the highest priority are used for synthesizing the virtual view, while the other input views may be used for disocclusion filling. In the proposed inpainting step, the search of the nearest points is performed using a transverse equirectangular projection (the Cassini projection [5]).

In equirectangular projection, usually used for omnidirectional video, a sphere is mapped onto a cylinder that is tangential to points on a sphere that have the latitude equal to 0 degrees (Fig. 2a). In transverse projection, the cylinder on which the sphere is mapped is rotated by 90 degrees, so it is tangential to points that have longitude equal to 0 degrees (Fig. 5b). Such change of the equirectangular projection causes, that the search for nearest projected points can be now performed simply in rows of an image.

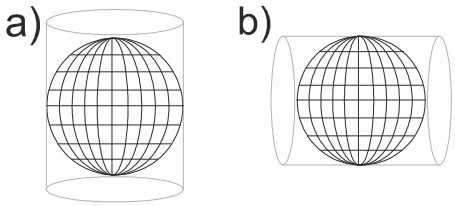

Fig. 5. Cylinders used in the projection of a sphere on a flat image in a) equirectangular projection and b) transverse equirectangular projection.

We propose the fast approximate reprojection of equirectangular image to transverse equirectangular image. First of all, length of all rows in an equirectangular image is changed to correspond to the circumference of the corresponding circle on a sphere (Fig. 6a). In the second step, all columns of such image are expanded (Fig. 6b), to be of the same length (Fig. 6c).

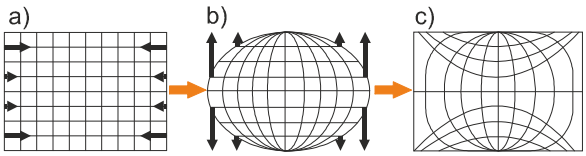

Fig. 6. Fast reprojection of an equirectangular image (a) to transverse equirectangular image (c). Black arrows show direction of change of size of respective rows and columns of images

.

After the two closest points are found in the transverse image, in order to perform the inpainting of a disoccluded point, we use the color, the depth, and the position of two closest projected points. If the depth values of these points are similar, the color of disoccluded point is the average color of two nearest points, weighed by the distances to these points. In the case of a difference of depth greater than the set threshold, only the color of the further point is used.

4.2 Depth-based filtration of edges

This post-processing technique increases the objective quality of virtual view synthesis. After performing the view synthesis, spatial edges are identified using input depth map, i.e., places where there are high gradients in depth map are identified. In those places, the synthesized view is filtered (blurred) in order to avoid sharp, unnatural edges.

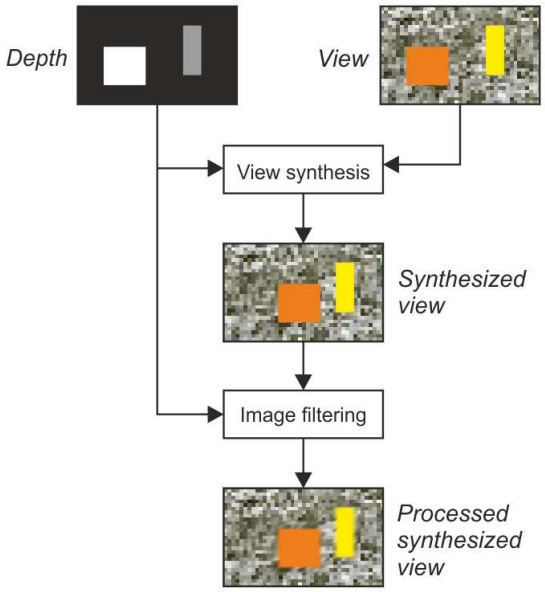

Fig. 7. The method of blurring the edges of spatial objects in a synthesized view.

5 Technical description of the decoder

The decoder consists of multiple HEVC decoders and view synthesis module.

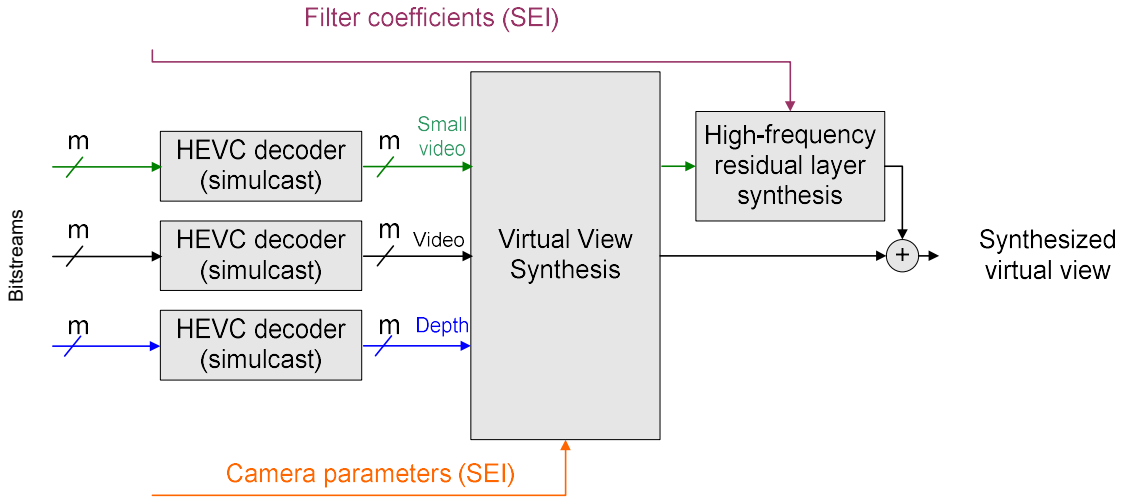

Fig. 8. The block scheme of the decoder.

It is assumed that there are *m* streams transmitted:

- Video (video base layer),
- Depth,
- Residual video layer, i.e., a stream of parameters representing the noise-like components from the input video.

The transmitted videos may be in the following formats:

- Flat (rectangular) videos
- Equirectangular projection (ERP) videos

The resolutions of each of these streams may be different, e.g.

- depth may be decimated with respect to the video (x2 or x4)
- spectral density of energy of residual layer may be vastly smaller (e.g. decimated by factor of 30 or sent as single value if the density of energy is uniform).

The decoded videos are fed to the view synthesis software which synthesizes the asked virtual view. It may be:

- one of the views specified in the input representation
- any other position of virtual view.

The virtual view synthesis must be capable of ignoring regions of the input videos which are not usable. E.g. in the case of supplementary views, only disoccluded regions are transmitted and the rest of the video is inpainted with gray color.
The output of the view synthesis are:

- synthesized video at the position of the virtual view,
- synthesized spectral density energy of the residual layer.

After view synthesis, high-frequency residual layer is generated synthetically (Fig. 9) basing on:

- spectral density of energy, synthesized to the position of desired virtual view
- spectral envelope, reconstructed basing on IIR filter coefficients sent in LAR format.

Afterwards, high-frequency residual layer is added the content of the synthesized view.

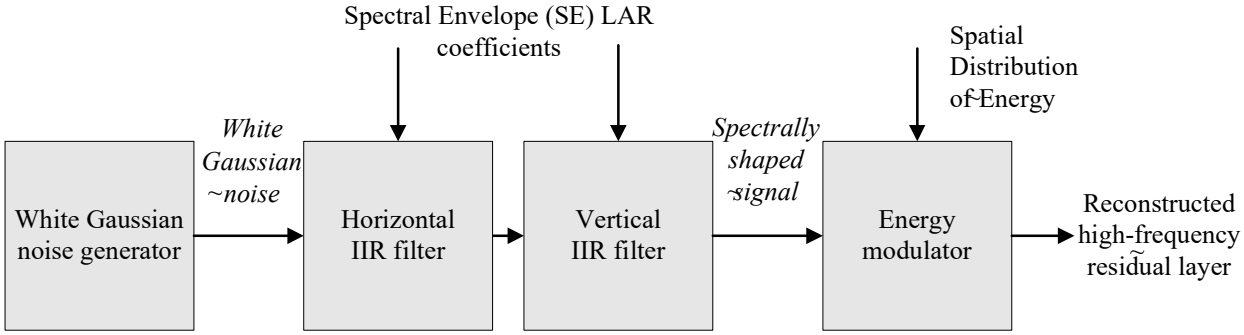

Fig. 9. Reconstruction of the high-frequency component in the decoder.

6 Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00207, Immersive Media Research Laboratory).

7 IPR statement

Poznań University of Technology and Electronics and Telecommunications Research Institute may have IPR relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the UTI-T/ITU-R/ISO/IEC patent statement and licensing declaration form).

8 References

[1] "Call for Proposals on 3DoF+ Visual" ISO/IEC JTC1/SC29/WG11 MPEG/N18145, January 2019, Marrakesh, MA.

[2] http://avisynth.org.ru/mvtools/mvtools.html

[3] A. Dziembowski, A. Grzelka, D. Mieloch, O. Stankiewicz, K. Wegner, M. Domański, "Multiview Synthesis – improved view synthesis for virtual navigation," 32nd Picture Coding Symposium, PCS 2016, Nürnberg, Germany, 4-7.12.2016.

[4] A. Dziembowski, O. Stankiewicz, "[MPEG-I Visual] Fast color correction technique for view synthesis," ISO/IEC JTC1/SC29 WG11 MPEG 123th meeting, Ljubljana, July 2018, Doc. M43694.

[5] J. Snyder, P. Voxland, "An album of map projections", US Government Printing Office, Washington, 1989.