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

ISO/IEC JTC1/SC29/WG11 MPEG2018/m42941
July 2018, Ljubljana, Slovenia

Title	**Integration of multiple input views into the View Synthesis Reference Software**
Status	**Input**

Source **Poznań University of Technology**

Authors **Adrian Dziembowski,**
Dawid Mieloch,
Krzysztof Wegner (kwegner@multimedia.edu.pl),
Olgierd Stankiewicz,
Marek Domański

1. Introduction

This document reports integration of recently proposed extension of View Synthesis Reference Software [m42526], which allows to synthesize the virtual view based on multiple input views. Such approach allows to significantly increase the quality of synthesized views, especially for the scenes with significant number of occlusions.For two input views candidate version 5.0 of View Synthesis Reference Software works exactly the same as version 4.2.

Other minor additions described in this document are:

- Modified way of calculation blending weights for different input views.
- Blending weights compatibility mode for two input views.
- Changing the depth-based view blending via a parameter in the configuration file (ViewBlending = 2).
  - Compilation project for linux
  - Bug fixes in NICT Inpainting
  -

2. Introduced extensions

2.1. Support for multiple views

2.1.1. Introduction

Algorithm implemented in View Synthesis Reference Software version 4.2 composed of two processing paths, one for left input view, and one for right input view, followed by virtual view blending and inpainting.

In the candidate version 5.0 of View Synthesis Reference Software implementation have been refactored to allow allocation of unlimited processing path, one per input views. Exact number of input views can be specified in configuration file.

The scheme of VSRS with multiple input views is presented in Fig. 1.

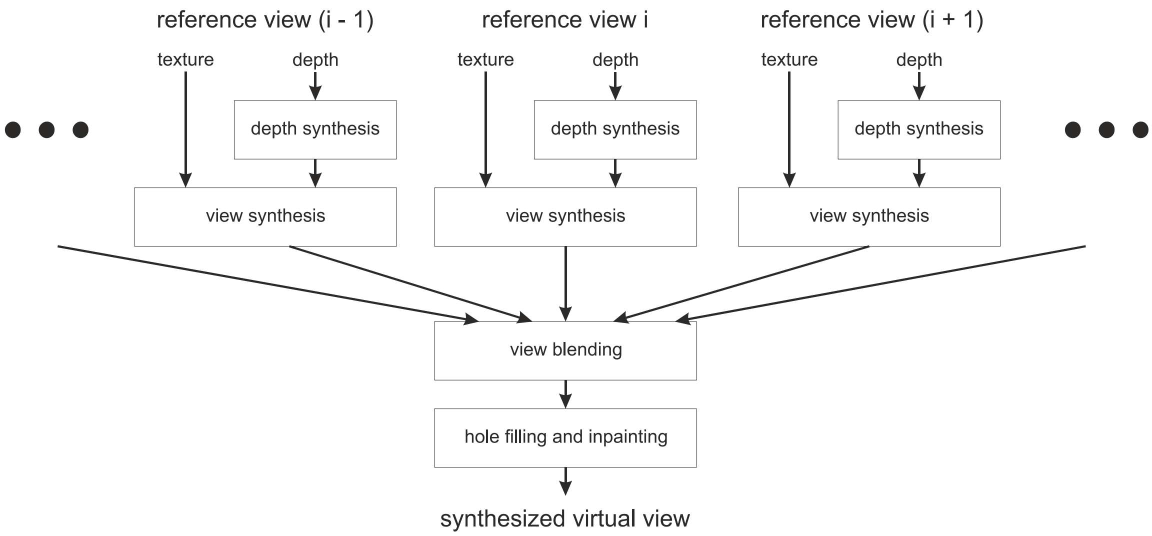

Fig. 1. View Synthesis Reference Software with multiple input views

2.1.2. View naming convention

Proposed extension allows to use practically unlimited number of input views and corresponding depth maps. Instead using names Left and Right consecutive camera names, input views, depth maps and nearest/farthest depth values are numbered, e.g.:

```
CameraName0
CameraName1
CameraName2
…

ViewImageName 0
ViewImageName 1
ViewImageName 2
…

DepthMapName 0
DepthMapName 1
DepthMapName 2
…

NearestDepthValue0
NearestDepthValue1
NearestDepthValue2
…

FarthestDepthValue0
```

```
    FarthestDepthValue1
    FarthestDepthValue2
    …
```

Due to memory savings, there is a compilation flag (MULTIREFERENCE_LENGTH), which allows to set the maximum number of input views allowed.

Exemplary configuration file:
New configuration parameters have been highlighted in red. Configuration parameters with change names are marked blue.

```
#================ Input Parameters ================
DepthType                      1                                  # 0...Depth from camera,
1...Depth from the origin of 3D space
SourceWidth                    1280                        # Input frame width
SourceHeight                   768                         # Input frame height
StartFrame                     0                           # Starting frame
TotalNumberOfFrames            1                           # Total number of input frames
NumberOfReference              3                           # Number of reference views
#------------------------------------------------------------------------------------------
NearestDepthValue0             0.2
FarthestDepthValue0            700
NearestDepthValue1             0.2
FarthestDepthValue1            700
NearestDepthValue2             0.2
FarthestDepthValue2            700
#------------------------------------------------------------------------------------------
CameraParameterFile            BigBuckBunny_Flowers_Arc_Wegner.txt       # Name of text
file which includes real and virtual camera parameters

CameraName0                   param_ptm19                  # Name of real first camera
CameraName1                   param_ptm32                  # Name of real second camera
CameraName2                   param_ptm45                  # Name of real third camera

ViewImageName0                BBB_Flowers_noblur_1280x768_24Hz_8b_P420_cam0019.yuv
ViewImageName1                BBB_Flowers_noblur_1280x768_24Hz_8b_P420_cam0032.yuv
ViewImageName2                BBB_Flowers_noblur_1280x768_24Hz_8b_P420_cam0045.yuv

DepthMapName0                 BBB_Flowers_1280x768_Depth_8bps_cf420_cam0019.yuv
DepthMapName1                 BBB_Flowers_1280x768_Depth_8bps_cf420_cam0032.yuv
DepthMapName2                 BBB_Flowers_1280x768_Depth_8bps_cf420_cam0045.yuv

VirtualNearestDepthValue       0.2
VirtualFarthestDepthValue      700
VirtualCameraName             param_ptm31                  # Name of virtual camera
OutputVirtualViewImageName    BBB_Flower_synth_1280x768_24Hz_8b_P420_Blender.yuv
# Name of output virtual view video
ColorSpace                     0                                              # 0...YUV,
1...RGB
Precision                      4                                              #
1...Integer-pel, 2...Half-pel, 4...Quater-pel
Filter                         0                                              # 0...(Bi)-
linear, 1...(Bi)-Cubic, 2...MPEG-4 AVC

BoundaryNoiseRemoval           0                                             # Boundary
Noise Removal: Updated By GIST
```

```
SynthesisMode                  0                                              #
0...General, 1...1D parallel, 2...360 degree

DepthBlendDifference           1
#---- General mode ------
ViewBlending                   2                                              # 0...Blend
left and right images, 1...Not Blend, 2...Depth based view blending

IvsrsInpaint                   1                                              #
0...Conventional, 1...NICT improved one

#---- 1D mode      ------
#---- In this example, all parameters below are commented and default values will be taken
#SplattingOption       2                 # 0: disable; 1: Enable for all pixels; 2: Enable
only for boundary pixels. Default: 2
#BoundaryGrowth        40                # A parameter to enlarge the boundary area with
SplattingOption = 2. Default: 40
#MergingOption         2                 # 0: Z-buffer only; 1: Averaging only; 2: Adaptive
merging using Z-buffer and averaging. Default: 2
#DepthThreshold        75                # A threshold is only used with MergingOption = 2.
Range: 0 ~ 255. Default: 75
#HoleCountThreshold    30                # A threshold is only used with MergingOption = 2.
Range: 0 ~ 49. Default: 30
```

2.1.3. Modified blending weight calculation

So far the blending weights for two input views were calculated using: $w_L = 1 - d_R$ and $w_R = 1 - d_L$, where w_L and w_R are the weights for left and right input view; d_L and d_R are the distances between the virtual camera position and, respectively, left and right real camera position. This method does not work for multiple input views.

In integrated multiview extension, the blending weight for a i-th input view is calculated as $w_i = \frac{1}{d_i}$, where d_i is the distance between the virtual view position and i-th input view camera position. All weight w_i are normalized so $\sum_i w_i = 1$.

2.1.4. Blending weights compatibility mode for two views

In order to maintain the compatibility with previous versions, there is a possibility to switch on during compilation of a software the two-view-blending-weight compatibility mode. In compatibility mode previous blending weights calculation is used when only two input views are provided in configuration file. When only two input view rendering is requested for input view 0 the blending weight is $w_0 = 1 - d_1$, and for input view 1: $w_1 = 1 - d_0$.

2.1.5. Multiview depth based view blending

So far the in depth-based view blending stage [m37232] for two input views, selects pixel color from two virtual view candidates rendered out from two input views based on depth value. Color of nearest pixel based on depth value was selected, or in case of similar depth value (where similarity can be controlled from configuration file) both colors from left and right based virtual view have been blended together according to blending weight (see sec 2.1.4).

In case of multiple virtual view candidates rendered from multiple input views, principle doesn't change, so color of the pixels that are closest to the virtual view camera position are blended together, all other are discarded. This simple principle have been implemented as follows.

First for a given pixel position considered, closest depth value (largest disparity) d_{max} along all depth maps of virtual view candidate have been selected.

Virtual view candidate which depth value d_i for processed pixel position is farther (smaller) than closest depth value (largest disparity) d_{max} selected in previous step by provided in configuration file threshold is discarded and will not be used in actual color blending.

Color of the processed pixel position of all virtual view candidate remaining are blended together with blending weights (see sec 2.1.3).

2.1.6. Depth based view blending compatibility mode for two views

In order to maintain the compatibility with previous versions, there is a possibility to switch on during compilation of a software the two-view-depth-based-view-blending compatibility mode. In compatibility mode previous depth based view blending routine is executed

2.1.7. Implementation details

Multi-input-view support have been implemented on top of VSRS 4.2. All new introduced source code have been surrounded by `MULTIREFERENCE` compilation frag allowing new feature switch on and off, during compilation.

Maximum of allowed input view can be limited during compilation by setting `MULTIREFERENCE_LENGTH` flag. By default number of input view is limited to 10.

Two-view-blending-weight compatibility mode can be switch on/off during compilation by `MULTIREFERENCE_TWO_VIEW_WEIGHT_COMPABILITY_MODE` compilation flag.

Two-view-depth-based-view-blending compatibility mode can be switch on/off during compilation by `MULTIREFERENCE_TWO_VIEW_DEPTH_BASED_VIEW_BLENDING_COMPABILITY_MODE` compilation frag

2.2. View blending mode selection from configuration file

VSRS 4.1 allowing switch on and off blending stage from configuration file. m37232 have introduced depth-based view blending, but it was implemented only as compilation flag, effectively disallowing usage of older view blending mode. In proposed candidate version 5.0 view blending stage have been refactored, allowing selection of view blending method from configuration file. Moreover new code structure allow easier implementation of new view blending methods in future.

The following configuration parameters have been reused for view blending selection:

```
ViewBlending    2  # 0...Blend left and right, 1...Not Blend, 2...Depth based view blending
```

`ViewBlending`

Unsigned int (0, 1 or 2), default: 0

Specifies the mode of view blending. 0 means simple blending of left and right image, 1 means no blending, 2 means the depth-based view blending.

2.3. Independent z range for virtual view

Each input view have z-range specified in configuration file by two configuration parameters NearestDepthValue and FartherDepthValue. This value restricts range of distances (z-values) stored in depth map associated with given input view. When virtual view is being rendered distances to a objects in the scene seen from virtual view camera position needs to be stored in depth map of newly rendered virtual view. For normalizing distances (z-value) to normalized disparity which is a used depth map format (see m40019) Nearest- and FartherDepthValue for virtual view are needed. In version 4.2 those values are copied form one of the input views. Such an approach have no problem when only in-plain rendering is done. But when one wants to render virtual view from virtual camera placed further away from the scene this leads to serious problems. In such condition distances seen form such virtual view are larger than those seen form input view, and the z-range of the depth map associated with virtual view is not sufficiently large to store the z-value correctly leading to virtual view distortion.

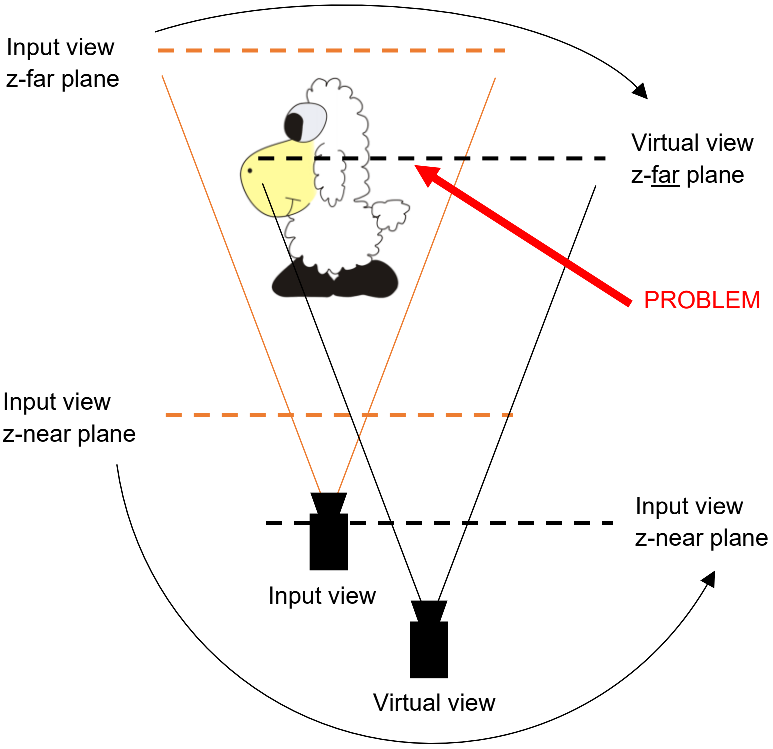

Figure 1. Ilustration of a problem with virtual view z-range.

In the candidate version z-range for virtual view can be specified independent from z-ranges of each input views. New **`VirtualNearestDepthValue`** and **`VirtualFartherDepthValue`** configuration parameters have been introduced to cope with described problem. By setting appropriate values for z-range of virtual view problem can be avoided.

3. Results

Proposed candidate version 5.0 have been compared against View Synthesis Reference Software version 4.2. Both software's have been compiled in Visual Studio 2017 against OpenCV 3.4.0 under 64-bit windows.

Experiments have been done according to Exploration Experiments for MPEG-I: Windowed-6DoF [w17609] for ULB Unicorn Plan A' data set. Latest configuration files for VSRS 4.2 been download from MPEG content repository [N17609]. Configuration files for candidate VSRS 5.0 have been created based on configuration files for VSRS 4.2, and are attached to the contribution.

Table 1. Quality of virtual view rendered from two input views obtain for ULB Unicorn plane A' dataset. Configuration of the render according to [N17609].

Rendered View Position	Input View Position	Input View Position	PSNR [dB]		Gain [dB]
			VSRS 4.2	VSRS 5.0	
5003070370	5003070404	5003410370	28,35	29,41	1,06
5003070404	5003070370	5003070438	29,12	30,34	1,22
5003070438	5003070404	5003070472	29,32	30,56	1,24
5003070472	5003070438	5003070506	29,31	30,81	1,51
5003070506	5003070472	5003410506	28,55	29,30	0,76
5003410370	5003070370	5003750370	27,99	29,54	1,55
5003410404	5003070370	5003750438	25,79	27,04	1,25
5003410404	5003070404	5003750404	27,55	29,07	1,52
5003410404	5003410370	5003410438	28,55	29,81	1,26
5003410438	5003070404	5003750472	25,94	27,12	1,19
5003410438	5003070438	5003750438	27,37	28,88	1,51
5003410438	5003410404	5003410472	28,68	30,13	1,45
5003410472	5003070438	5003750506	25,96	27,10	1,14
5003410472	5003070472	5003750472	27,36	28,87	1,51
5003410472	5003410438	5003410506	28,59	29,87	1,28
5003410506	5003070506	5003750506	27,91	29,41	1,50
5003750370	5003410370	5004090370	28,47	29,98	1,52
5003750404	5003410370	5004090438	25,71	27,06	1,36
5003750404	5003410404	5004090404	27,88	29,55	1,67
5003750404	5003750370	5003750438	28,54	30,03	1,50
5003750438	5003410404	5004090472	25,70	27,03	1,33
5003750438	5003410438	5004090438	27,70	29,33	1,63
5003750438	5003750404	5003750472	29,29	30,42	1,13
5003750472	5003410438	5004090506	26,02	27,16	1,14
5003750472	5003410472	5004090472	27,74	29,32	1,58
5003750472	5003750438	5003750506	29,28	30,22	0,95
5003750506	5003410506	5004090506	28,57	30,14	1,57
5004090370	5003750370	5004430370	26,84	28,33	1,49
5004090404	5003750370	5004430438	25,81	26,91	1,11
5004090404	5003750404	5004430404	26,66	28,18	1,51
5004090404	5004090370	5004090438	28,10	30,06	1,96

5004090438	5003750404	5004430472	25,54	26,61	1,08
5004090438	5003750438	5004430438	26,88	28,31	1,44
5004090438	5004090404	5004090472	28,86	30,20	1,35
5004090472	5003750438	5004430506	26,04	27,08	1,05
5004090472	5003750472	5004430472	26,76	28,30	1,55
5004090472	5004090438	5004090506	28,98	30,09	1,11
5004090506	5003750506	5004430506	27,91	29,22	1,32
5004430370	5004430404	5004090370	28,47	29,16	0,69
5004430404	5004430370	5004430438	29,68	31,19	1,51
5004430438	5004430404	5004430472	29,89	31,14	1,24
5004430472	5004430438	5004430506	29,58	30,92	1,33
5004430506	5004430472	5004090506	28,38	29,68	1,30
			Average Gain		**1,33**

Without any modification to the configuration files new candidate VSRS 5.0 provide 1.33 dB gain on average.

4. Conclusions

New candidate VSRS version have been prepared. It allows for unlimited number of input views to be used for rendering requested virtual view. Initial evidence of improvement have been provided. Never the less further study of the performance of new candidate VSRS version is needed.