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/m46124
January 2019, Marrakesh, Morocco

Source	**Poznań University of Technology, Poznań, Poland** **Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea**
Status	**Information**
Title	**Depth estimation for omnidirectional video with various representations**
Author	**Marek Domański*, Olgierd Stankiewicz*, Gwangsoon Lee**, Jeongil Seo**, Adrian Dziembowski*, Dawid Mieloch*, Krzysztof Wegner*** * - Poznań University of Technology, **- Electronics and Telecommunications Research Institute
Contact	marek.domanski@put.poznan.pl, olgierd.stankiewicz@put.poznan.pl

Abstract

Omnidirectional video formats are currently considered within MPEG in the context of prospective and Omnidirectional 6DoF/3DoF+ video technology. Unfortunately, the current version of Depth Estimation Reference Software does not allow depth estimation from video acquired by multiple omnidirectional cameras, needed to create multi-point 6DoF/3DoF+ scene representation. In this document we present novel depth estimation technique and software developed by Poznań University of Technology (PUT) and Electronics and Telecommunications Research Institute (ETRI), which addresses these deficiencies.

1 Introduction

One of the subjects considered in the current MPEG-I activities in the context of prospective 6DoF/3DoF+ video technology are omnidirectional video formats. Acquisition of video with an omnidirectional camera or with several omnidirectional cameras positioned in distinct locations seems to be very promising way to capture real, natural 3D content. It is therefore important to develop tools allowing further research on 360 video, starting with 3DoF+ video. Thus, the motivation for this work is to develop depth estimation software allowing usage of input omnidirectional videos, e.g. in equirectangular (ERP) video format.

We consider a merged use of perspective and omnidirectional cameras. Unfortunately, DERS (Depth Estimation Reference Software), developed by the MPEG community is a very complex software, therefore the assurance of the correct interpretation of omnidirectional cameras video would require to change the code of DERS to a significant extent. Moreover, the current direction of changes in the DERS software (versions 6.0 – 6.2) show that this framework is being adopted for lightfield systems. Therefore Poznań University of Technology (PUT) and Electronics and Telecommunications Research Institute (ETRI) have decided to develop a new, dedicated software.

Also, in order to reduce the high complexity of the depth estimation we proposed use of the parallelization of the computations. The prepared depth technique is not based on any assumption about a number, a positioning, and a type of used cameras.

2 Algorithm

The algorithm is based on the earlier results obtained by Poznan University of Technology, Chair of Multimedia Telecommunications and Microelectronics. These results have been already published in [2] and [4]. Recently, these results have been used and extended for a technique for omnidirectional video

The algorithm utilizes superpixels in order to decrease the complexity of the depth estimation. The estimation is based on a cost function minimization using graph cut algorithm. The cost function is based on two components: the intra-view discontinuity cost $V_{s,t}$ and the inter-view matching cost $M_{s,s'}$, responsible for the inter-view consistency of depth maps:

$$E(\underline{\mathrm{d}}) = \sum_{c \in \mathrm{C}} \left\{ \sum_{c' \in \mathrm{D}} \sum_{s \in \mathrm{S}} M_{s,s'}(d_s) + \sum_{s \in \mathrm{S}} \sum_{t \in \mathrm{T}} V_{s,t}(d_s, d_t) \right\},$$

where:

$\underline{\mathrm{d}}$	–	vector of depth values for all segments in all views,
C	–	set of views,
c	–	view used in the estimation,
D	–	set of views neighboring to the view c,
c'	–	view neighboring to the view c,
S	–	set of segments of the view c,
s	–	segment in the view c,
d_s	–	currently considered depth of the segment s, d_s is a component of $\underline{\mathrm{d}}$,
s'	–	segment in the view c', which corresponds to the segment s in the view c for the currently considered depth d_s ,
$M_{s,s'}$	–	inter-view matching cost between segments s and s',
T	–	set of segments neighboring to the segment s,
t	–	segment neighboring to the segment s,
$V_{s,t}$	–	intra-view discontinuity cost between segments s and t,
d_t	–	currently considered depth of the segment t, d_t is a component of $\underline{\mathrm{d}}$.

Unlike in typical formulations of graph-based depth estimation, in which each node in a constructed graph represents one point of an input view, in the proposed method each node of a graph corresponds to one segment (Fig. 1). Nodes are connected with each other by two types of links that represent the intra-view discontinuity and inter-view matching costs.

In order to achieve inter-view consistency of estimated depth maps, the matching cost is not calculated independently for each single view. Instead, the conventional matching cost is replaced with the inter-view matching cost $M_{s,s'}(d_s)$ that is calculated in the pixel-domain in a user-defined window around the center of a segment and the corresponding point in a neighboring view. Such definition of the inter-view matching cost does not require segmentation to be inter-view consistent in neighboring views, therefore, segmentation can be performed independently for each view.

In depth estimation for omnidirectional cameras the possibility of the use of both equirectangular and perspective views is included in the 3D transform of a point to the another view. The use of typical 3D transform would result in the incorrect estimation of depth maps. A point from an image acquired using normal (perspective) camera has the corresponding point in neighboring equirectangular view not on an epipolar line, as in the traditional depth estimation.

The intra-view discontinuity cost is calculated between all neighboring segments within the same view. In the proposed method the smoothing coefficient is not fixed for all segments, instead, the smoothing coefficient is calculated using a similarity of two neighbouring segments an initial smoothing coefficient.

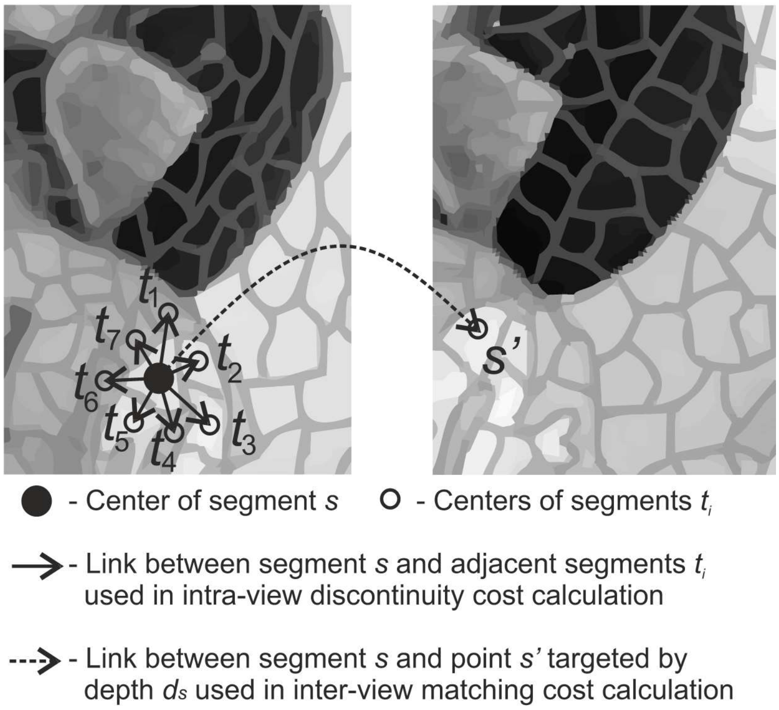

Fig. 1. The intra-view discontinuity cost and the inter-view matching cost for a segment s for depth estimation performed for 2 views.

In the proposed method, several simultaneous depth estimation processes can be performed. Each depth estimation process performs graph cut optimizations for different sets of depth levels. Depth maps with the reduced number of depth levels that were calculated by different threads have to be merged into one depth map. The merging process is performed in a similar way as depth estimation, using the optimization of the cost function, but only two levels of depth are considered for each segment in one cycle of merging – the depths of a segment calculated by two different threads.

3 Experiments

We have tested experimentally the developed depth estimation software with the use of set of computer-generated omnidirectional dataset provided to MPEG on the last meeting [1]: House 360, People 360, Space 360, Blocks 360.

Usage of computer-generated synthetic content, allowed us to compare the quality of the estimated depth with ground-truth data. In particular, we have used both sets of depth maps (ground truth and depth maps estimated with the provided software) to synthesize sweep sequences. Then, the quality of the sweep attained on the estimated depth was assessed by the means of PSNR.

The results are presented in Figs. 2-3 and in Table 1.

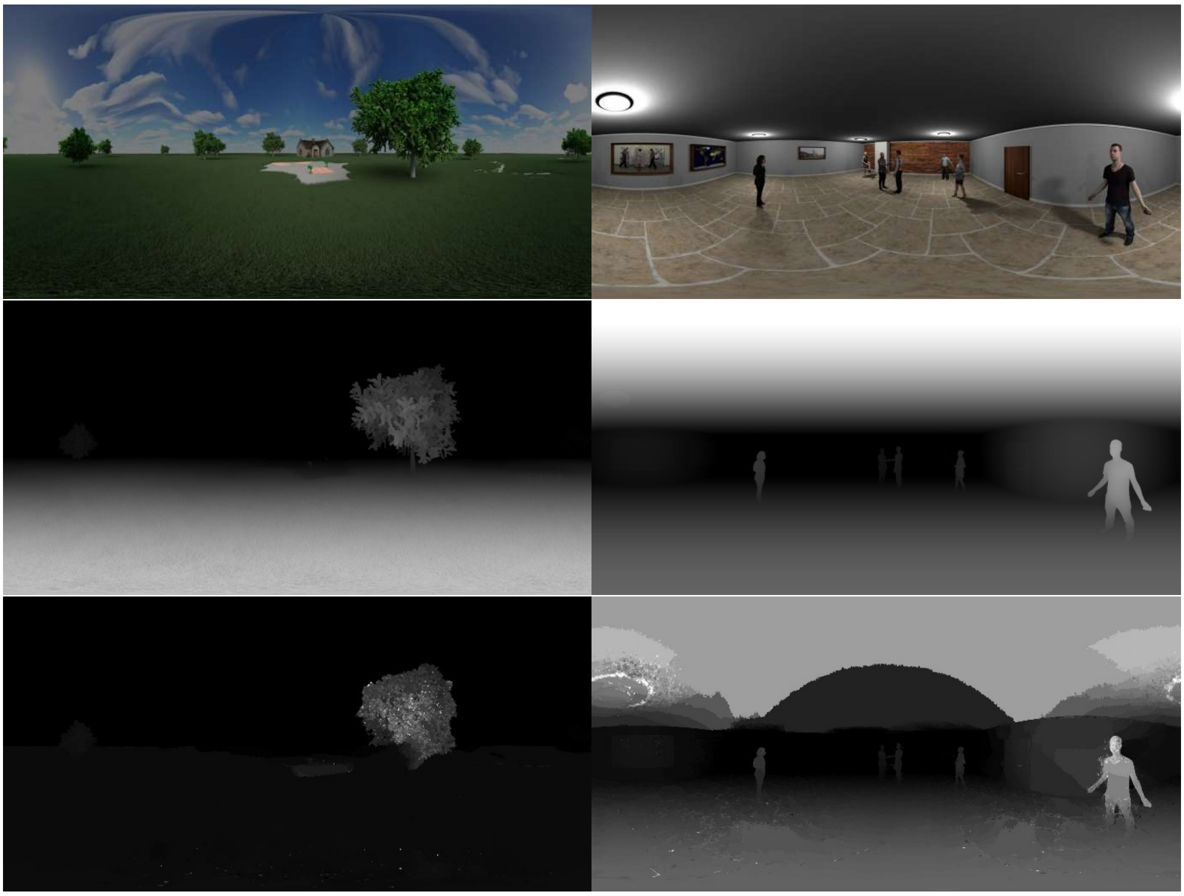

Fig 2. Results for “House 360” (left) and “People 360” datasets.
The omnidirectional image (top row), ground-truth depth map (middle row) and the estimated depth (bottom row).

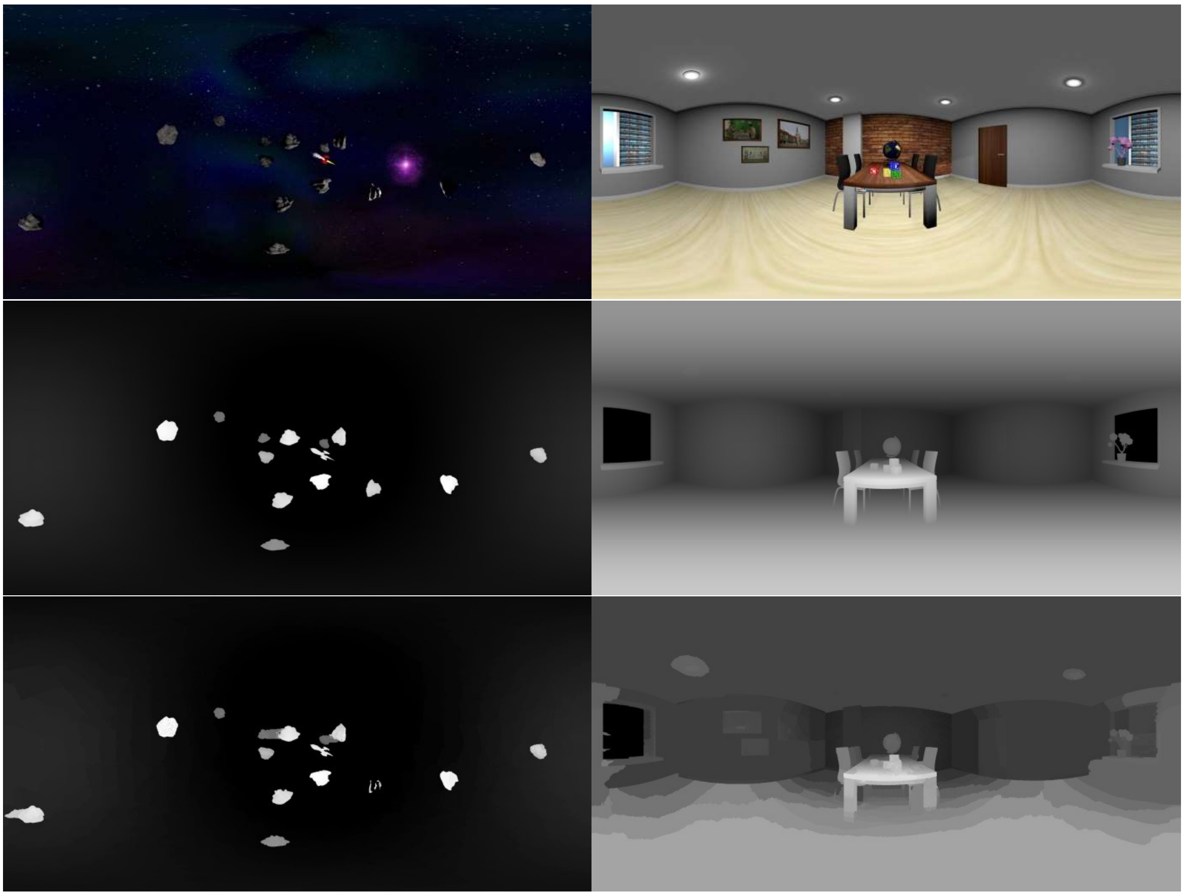

Fig 3. Results for “Space 360” (left) and “Blocks 360” datasets.
The omnidirectional image (top row), ground-truth depth map (middle row) and the estimated depth (bottom row).

Table 1. Quality evaluation of the depth estimation by means of sweep view synthesis.

Sequence	**PSNR [dB]**
Blocks 360	23.92
House 360	29.47
People 360	26.82
Space 360	30.04
Average	**27.56**

4 Availability of the software

The software remains the property of Poznan University of Technology and Telecommunications Research Institute (ETRI). Usage of its executable form is licensed for free use within ISO/IEC JTC1/SC29/WG11 (MPEG) for the purposes of research and the development of standards. Any other use is prohibited unless an explicit permission is given by Poznań University of Technology, Chair of Multimedia Telecommunications and Microelectronics and/or Electronics and Telecommunications Research Institute and Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea.

Acknowledgements are appreciated if the software was used in research and are required if the software is to be used in publications. The acknowledgement should use the reference to this document.

The software (its executable form) can be made available upon request.

5 Use of developed software

The framework includes all necessary functions used in order to estimate depth maps for perspective and omnidirectional views. The software is configured by the configuration file, which uses following parameters:

- NumOfThreads - number of threads used for the parallelization
- StartFrame - number of frame of sequence used for the estimation
- TotalNumberOfFrames - number of frames used
- FileCameraParameter - name of the file containing VSRS-style camera parameters
- ChrominanceFormat - used format of chroma subsampling
- NearestZValue - nearest value of z for the scene
- FarthestZValue - farthest value of z for the scene
- NumberOfZSteps - number of z steps (depth levels)
- MatchNeighbors - number of neighboring cameras used in calculation of inter-view matching cost
- MatchThresh - value K (typically in our research equal to 30)
- Matcher - type of used matcher (Pixel or Block)
- MatchingBlockSize - size of block used in matching
- SmoothingCoefficient - initial smoothing coefficient
- NumOfCycles - number of optimization cycles
- NameOfCamera0 - name of camera (the same as in the camera parameter file)
- InputView0 - the path and name of file containing input view for camera 0
- ViewType0 - the type of view 0 (Perspective or Omnidirectional)
- ViewWidth0 - width of view 0
- ViewHeight0 - height of view 0
- ViewNumOfSuperpixels0 - number of superpixels for view 0
- OutputDepthMap0 - the path and name of file containing depth for camera 0
- SuperpixelSegmentationType - type of used segmentation (currently SLIC)
- SuperpixelColorCoeff - the weight of the color cost used in superpixel segmentation (typically in our research equal to 5)

The developed depth estimation software estimates depth map for mixture of omnidirectional and perspective cameras. The configuration file for the scenario of using 2 omnidirectional cameras in the depth estimation:

```
#=============== INPUT PARAMETERS ===============
NumOfThreads                    1
StartFrame                      0
TotalNumberOfFrames             1

FileCameraParameter             B360p.txt
ChrominanceFormat               420
NearestZValue                   2.5
FarthestZValue                  100
NumberOfZSteps                  250
MatchNeighbors                  2
MatchThresh                     30
Matcher                         Block
MatchingBlockSize               3
SmoothingCoefficient            0.1
NumOfCycles                     1

NameOfCamera0                   param_camL_4k
InputView0                      ImageL0047.png_4096x2048.yuv
ViewType0                       Omnidirectional
ViewWidth0                      4096
ViewHeight0                     2048
ViewNumOfSuperpixels0           1000000
OutputDepthMap0                 B360_L360.png_cf400_16bps_4096x2048.yuv

NameOfCamera1                   param_camC_4k
InputView1                      ImageC0047.png_4096x2048.yuv
ViewType1                       Omnidirectional
ViewWidth1                      4096
ViewHeight1                     2048
ViewNumOfSuperpixels1           1000000
OutputDepthMap1                 B360_C360.png_cf400_16bps_4096x2048.yuv

NameOfCamera2                   param_camR_4k
InputView2                      ImageR0047.png_4096x2048.yuv
ViewType2                       Omnidirectional
ViewWidth2                      4096
ViewHeight2                     2048
ViewNumOfSuperpixels2           1000000
OutputDepthMap2                 B360_R360.png_cf400_16bps_4096x2048.yuv

#=============== SEGMENTATION ===============

SuperpixelSegmentationType      SNIC
SuperpixelColorCoeff            5
```

The configuration file for the scenario of using the omnidirectional camera and the perspective camera:

```
#=============== INPUT PARAMETERS ===============
NumOfThreads                    1
StartFrame                      0
```

```
TotalNumberOfFrames                 1

FileCameraParameter                 B360p.txt
ChrominanceFormat                   420
NearestZValue                       2.5
FarthestZValue                      100
NumberOfZSteps                      250
MatchNeighbors                      2
MatchThresh                         30
Matcher                             Block
MatchingBlockSize                   3
SmoothingCoefficient                0.1
NumOfCycles                         1

NameOfCamera0                       param_camL_4k
InputView0                          B360_L.png_4096x2048.yuv
ViewType0                           Perspective
ViewWidth0                          4096
ViewHeight0                         2048
ViewNumOfSuperpixels0               100000
OutputDepthMap0                     B360_L.png_cf400_16bps_4096x2048.yuv

NameOfCamera1                       param_camC360_4k
InputView1                          B360_C360.png_4096x2048.yuv
ViewType1                           Omnidirectional
ViewWidth1                          4096
ViewHeight1                         2048
ViewNumOfSuperpixels1               500000
OutputDepthMap1                     B360_O.png_cf400_16bps_4096x2048.yuv

NameOfCamera2                       param_camR_4k
InputView2                          B360_R.png_4096x2048.yuv
ViewType2                           Perspective
ViewWidth2                          4096
ViewHeight2                         2048
ViewNumOfSuperpixels2               100000
OutputDepthMap2                     B360_R.png_cf400_16bps_4096x2048.yuv

#=============== SEGMENTATION ===============

SuperpixelSegmentationType          SNIC
SuperpixelColorCoeff                5
```

6 Conclusion

The document provides a brief description of the technique and software for depth estimation from omnidirectional video, in particular for 3DoF+ and beyond. Software (executable file) can be provided for the MPEG-I activities.

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).

References

[1] Olgierd Stankiewicz, Krzysztof Wegner, Adrian Dziembowski, Mateusz Lorkiewicz, Gwangsoon Lee, Jeongil Seo, Marek Domański, “Proposed test materials for 3DoF+ or Omnidirectional 6DoF”, ISO/IEC JTC1/SC29/WG11 MPEG2015, M44461, October 2018, Macao, China.

[2] D. Mieloch, A. Dziembowski, A. Grzelka, O. Stankiewicz and M. Domański, “Graph-based multiview depth estimation using segmentation,” in 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, pp. 217–222.

[3] K. Wegner, O. Stankiewicz, T. Grajek, M. Domański, “Depth estimation from circular projection of 360 degree 3D video”, Doc. ISO/IEC JTC1/SC29/WG11 nr MPEG2017/m40596, April 2017, Hobart, Australia.

[4] Dawid Mieloch, “Depth Estimation in Free-Viewpoint Television,” PhD dissertation, Poznań University of Technology, Poznań, 2018.

[5] Krzysztof Wegner, Olgierd Stankiewicz, Tomasz Grajek, Marek Domański, “Depth estimation from stereoscopic 360-degree video”, IEEE Int. Conf. Image Proc. 2018, ICIP 2018, pp. 2945 – 2948.

[6] K. Wegner, O. Stankiewicz, M. Domański „Depth based view blending in View Synthesis Reference Software (VSRS)”, ISO/IEC JTC1/SC29/WG11 MPEG2015, M37232, Geneva, Switzerland, 15-23 October 2015.