ESTIMATION OF GLOBAL LUMINANCE FOR HOLOVIZIO 3D DISPLAY

Oleksii Doronin, Attila Barsi

Holografika Ltd, Budapest, Hungary

ABSTRACT

Current 3D (light field, LF) displays are capable to visualize the synthetic scenes simultaneously for different viewing positions. The user-perceived image changes smoothly within the field of view of such display. The global luminance level the of user-perceived image may change drastically with the viewing direction. This level is often used in image processing algorithms, such as tone mapping, in order to improve the image quality.

In this paper, we suggest a method of evaluating the global luminance level for a particular type of horizontal-parallax light field display. This method requires only a set of projector images needed for the actual rendering, and can work in real-time. We test the visual performance of this method on a synthetic scene, and as a part of the conventional spatially-uniform tone mapping algorithm.

Index Terms— Light field, 3D display, tone mapping, high dynamic range, HDR, rendering, visualization.

1. INTRODUCTION

Light field (LF) capturing technology has been actively developing since 1990s. Several solutions of 3D capturing systems (e.g., Lytro, Adobe light field camera, CAFADIS camera, etc.) are already available on the consumer market. Light field rendering systems, however, have just started entering the consumer market. Many solutions in this area are represented by the stereoscopic, multi-view, or eye-tracking systems, capable of rendering an interactive content for a limited number of users or possible user positions. However, the truly LF displaying technology aims to recreate the original flux of light coming from the captured or synthetic scene, which overall is similar to the effect of holography.

There can be several variations of LF displays. Some of important characteristics of them are: (a) field of view; (b) matrix resolution; (c) angular resolution; (d) type of parallax (full or horizontal-only). Full-parallax displays visualize a content in a way that the user-perceived image changes smoothly with the observer position, adapting accordingly. This creates an effect of observing the target scene from different angles, providing a great level of immersion. Full-parallax displays, however, are often limited either by a narrow field of view, or by static-only scenes, or both. Horizontal-parallax displays can often provide much higher field of view and resolution, and can easily render an interactive content. But this comes with the cost of adapting to the horizontal-only changes of user position.

Regardless of parallax type, it is often the case that the rendered scene should expose different levels of brightness for different observer positions. Measuring this level is crucial for the case of high dynamic range (HDR) scene visualization on low dynamic range (LDR) devices, in order to apply a proper tone mapping correction. This problem seems trivial for two following cases: (a) camera-captured scenes, with each camera representing a particular view; (b) multi-view displays with relatively small number of views. In both cases, the straightforward approach is to measure the global luminance level for each view individually, and apply needed version of tone mapping. However, such solution is not available for several types of 3D displays.

For example, HoloVizio display system consists of projectors, placed behind the diffuser surface with special holographic properties. The light field (LF) image is stored as a set of projector images. Projector images can also be understood as the collection of individual rays of light that come from the projector through the diffuser surface, hitting the horizontal line of possible user positions (*observer line*). Note that each individual pixel of a projector image makes its strongest contribution to a unique position on the observer line. In contrast to this, the conventional pinhole camera model constructs the whole image for just a single observer position. Thus, no single projector image can form a pinhole camera image, and vice versa.

This paper is structured as follows. Section 2 gives an overview of current state of research in LF visualization and tone mapping. Formal description of the global luminance estimation problem is placed in Section 3. Description of "brute-force" and "accumulative" methods that we suggest, can be found found in Sections 4 and 5 respectively. We include the details of GPU-optimized implementation of both methods in the corresponding subsections. Section 6 suggests possible extensions of this method to different types of light field displays. Section 7 provides the results of our simulation.

2. RELATED WORK

State-of-the-art of light field and holographic 3D displays is described by Yamaguchi [13]. HoloVizio system was described earlier by Balogh *et al.* [1]. The widely known lenticular light field display was discussed by van Berkel [11].

Eilertsen *et al.* [3] provides a good survey of the existing video tone mapping techniques. This work can also be considered as a good overview of tone mapping techniques in general, alongside with his previous work [4]. One of the most commonly used tone mapping methods is the one developed by Reinhard et al. [8]. This method models the behavior of the human visual system in terms of scaling the input luminance. To make the mentioned Reinhard's algorithm real-time, Slomp and Oliveira [10] suggest to use the *summed area tables* [2]. In his later work [9], Slomp *et al.* introduces the visual improvements of the tone mapping algorithm.

Tone mapping algorithm for multi-projection display was presented by Wang *et al.* [12]. Described display model, however, assumed that there is a one-to-one correspondence between individual projectors and possible user views, which is not the case for HoloVizio light field display. Mai *et al.* [6] provided subjective study of tone mapped images on 3D displays. Model for position-dependent luminance estimation for conventional 2D LED and LCD HDR displays, as well as optimization approach, was presented by Forchhammer and Mantel [5]. View-dependent method for rendering HDR images and spatially uniform tone mapping in VR using was described by Najaf-Zadeh *et al.* [7]. The introduced method, however, is applied only for a single viewer position (but different directions), and assumes that the input display image is rendered for that specific single viewer position.

According to our knowledge, no method of position-dependent luminance estimation for an arbitrary light field display had been published. Same holds true for tone mapping algorithms.

3. FORMALIZATION OF PROBLEM

Let us consider the case of a horizontal-parallax multi-projector light field display with semitransparent holographic diffuser surface (see Figure 1). We assume that the display surface is flat, and all projectors are placed behind it. By *observer space*, we mean the geometrical place of all possible positions of the user. For multi-view displays, the observer space is the discrete set of only a few positions. For the case of horizontal-parallax displays, we can assume that the observer space is represented by a straight horizontal line, which is parallel to the display surface (*observer line*).

We simulate a user at the particular position of the observer line by placing a virtual pinhole camera to that position (see Figure 1). The camera direction is perpendicular to the display surface; left/right and bottom/top edges of its image correspond to the left/right and top/bottom edges of the display. We repeat this for several positions on the observer line. As the result, we get the set of virtual pinhole cameras, each of them covering exactly the effective area of the display. Example images, rendered with this kind of camera setup, can be seen on Figure 2.

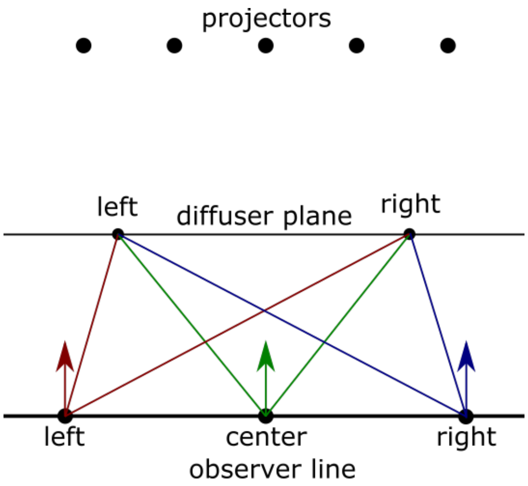


Fig. 1: Virtual cameras placement.

During the operation of the device, each projector receives its image as the input. An individual pixel of the projector image represents the ray of light that is spawned at the projector's position, then goes through the particular point of the display surface, and spreads its color contribution alongside the observer line. For each individual projector ray $\vec{r}$, there is a position X on the observer line that corresponds to its maximal contribution: $X = X(\vec{r})$. Our goal is to measure the global luminance level of the 2D image, that corresponds to the virtual pinhole camera placed at $X(\vec{r})$.

4. BRUTE-FORCE APPROACH

4.1. Few introductory notes

For tone mapping purposes, the global luminance level of a 2D image is usually estimated as the log-average:

$$\overline{L} := \exp\left[\frac{1}{\#\{p\}}\sum_{p}\log\left(\delta + L(p)\right)\right], \tag{1}$$

where sum is taken over all pixels p in the image, $L(p)$ is the luminance value of pixel p, $\#\{p\}$ is the total number of pixels, and δ is a user-defined small value to avoid taking logarithm of zero. In our applications, however, it is often the case that the projector images contain large completely black areas that should not even be visible. Or, for example, a synthetic scene may consist of an object "floating in the void", i.e., surrounded by completely black area that should not be taken into account during the rendering. This is why we use an alternative formula to compute the log-average luminance:

(a) From left. (b) From center. (c) From right.

Fig. 2: Scene “Pillars” in HDR.

$$\overline{L} := \exp\left[\frac{1}{\#\{p : L(p) > \varepsilon\}} \sum_{p:L(p)>\varepsilon} \log L(p)\right], \qquad (2)$$

where ε is a user-defined level of minimally acceptable luminance value, and average is calculated over all values with luminance bigger than ε.

We implement the averaging operation with the help of OpenGL mip-mapping functionality. We create a new square-shaped float-precision texture with width and height being equal to the smallest power of two that is bigger than width and height of the texture to be mip-mapped. The newly created texture has two channels (R and G): channel R contains $\log L(p)$ value if $L(p) > \varepsilon$ and 0 otherwise, and channel G contains 1 if $L(p) > \varepsilon$ and 0 otherwise. Pixel values outside the range of the initial texture are set to zero. After this texture has been initialized, we apply the mip-mapping operation until we reach the level of 1x1 texture. The final value of log-average luminance is calculated as the value from channel R of 1x1 texture divided by the value from channel G.

4.2. Method description

The main idea behind the brute-force method is to measure the global luminance level for the image of the pinhole camera on the observer line. First, we measure the global luminance level for dense-enough set of camera positions (samples) on the observer line. Next, we iterate over all pixels for all projector images, and assign to each pixel the value of global luminance level that corresponds to the position on the observer line that is hit by projector ray going through this pixel. The method is schematically described in Algorithm 1.

Algorithm 1 Brute-force method.

```
for all sampling points $s_j$ on observer line do
    put pinhole camera at $s_j$;
    generate camera image with given resolution;
    transform camera image into log-luminance texture;
    mip-map log-luminance texture;
    store value of 1x1 mip-map;
```

4.3. Implementation details

We implemented the brute-force method as the sequence of GLSL compute shaders. The following set of float-precision images was additionally created:

- *Lum3D* – projector images to store initial luminance.
- *GlobLum3D* – projector images to store the result.
- *MipMap2D* – square-shaped two-channel (RB) 2D image for mip-mapping.
- *Samples2D* – NumSamples-by-1 2D image to store the global luminance per each sample, where NumSamples is the total number of sampling points on the observer line.

The sequence of GLSL shaders is the following:

1. *PerceivedShader* – takes Lum3D, and generates perceived pinhole-camera image of user-specified resolution into MipMap2D, storing the log-luminance values.
2. *CopyShader* – calculates exponent of 1x1 mip-map level of MipMap2D, and stores this into Samples2D.
3. *GlobLumShader* – copies values of Samples2D into the corresponding pixels of GlobLum3D.

There are two main for-loops within a single render pass. The first loop iterates over all observer line positions, and calls PerceivedShader and CopyShader within a single iteration. After this, the global luminance levels of all samples are stored in Samples2D texture. The second loop iterates over all projector images, and calls GlobLumShader, which fills the final set of projector images containing the global luminance levels.

If the initial luminance in Lum3D texture is below the specified threshold ε, we write zero to both channels of MipMap2D texture. This texture is processed in the same way as the mip-map for conventional average luminance evaluation, described in Section 4.1.

5. ACCUMULATIVE APPROACH

5.1. Method description

In the accumulative method, we assume the predefined set of samples on the observer line. We iterate over all pixels in the input projector images, and for each pixel p we determine the nearest sample s_j to the observer line location $X(\vec{r})$ that corresponds to that pixel (we identify pixel p and the corresponding ray of light $\vec{r}$). We accumulate two values for each sample: the log-luminance of the corresponding pixels, and their number. Accumulative method is schematically described in Algorithm 2.

Algorithm 2 Accumulative method.

```
for all samples $s_j$ do
    $l_j \leftarrow 0$; $n_j \leftarrow 0$;
for all pixels in all projector images do
    generate ray $\vec{r}$ corresponding to current pixel;
    calculate position of maximal contribution $X(\vec{r})$;
    calculate index $j$ of closest to $X(\vec{r})$ sample;
    calculate log-luminance of current pixel;
    accumulate log-luminance $l_j$;
    increase (+1) number $n_j$;
for all samples $s_j$ do
    divide $l_j$ by $n_j$, and store result;
```

5.2. Implementation details

Analogously to the brute-force method, we implement the accumulative method as the sequence of GLSL compute shaders. We involve the following textures in this process:

- *Lum3D* – projector images with initial luminance.
- *GlobLum3D* – projector images for the result.
- *SamplesProjector2D* – NumSamples-by-1 two-channel (RB) 2D image to accumulate the log-luminance per samples; used individually for each projector image.
- *SamplesAll2D* – NumSamples-by-1 two-channel (RB) 2D image to accumulate the log-luminance per sample.
- *Samples2D* – NumSamples-by-1 2D image to store the resulting sampled global luminance.

The sequence of GLSL shaders is the following:

1. *AccumulateShader* – takes Lum3D, and accumulates log-luminance into SamplesProjector2D for a particular projector.
2. *SummarizeShader* – takes SamplesProjector2D for current projector, and adds it to SamplesAll2D.
3. *FinalizeShader* – takes SamplesAll2D, divides channel R by channel G, and writes the result into *Samples2D*.
4. *GlobLumShader* – copies the values of Samples2D into the corresponding pixels of GlobLum3D.

There are two main for-loops within a single render pass. First loop iterates over all projector images, and calls AccumulateShader and SummarizeShader for each particular projector. After this, SamplesAll2D texture is evaluated. Inbetween two loops, the FinalizeShader is called, which transforms SamplesAll2D texture into Samples2D. Second loops calls GlobLumShader, filling the final set of projector images with the evaluated global luminance values.

If the initial luminance in Lum3D is below ε, than the current pixel is ignored. Note that it is possible to improve FinalizeShader by adding a smoothing operation. E.g., instead of computing the global luminance level g_j for sample s_j as

$$g_j = \exp\left(\frac{l_j}{n_j}\right), \tag{3}$$

one can improve it by considering the values from adjacent samples:

$$g_j = \exp\left(\frac{l_j + 0.5l_{j-1} + 0.5l_{j+1}}{n_j + 0.5n_{j-1} + 0.5n_{j+1}}\right). \tag{4}$$

5.3. Difference to the brute-force method

When the projector ray hits the display surface, it is being spread across all the observer line, with intensity depending on deflection from the main ray direction. This fact is taken into account in the brute-force method, because this method recreates the actual pinhole-camera perceived images. However, in accumulative method, any projector ray contributes only to a single position on the observer line. This difference may cause estimation artifacts if the luminance level is drastically different for similar pixel positions of adjacent projectors.

It is possible to mitigate the mentioned drawback of the accumulating method by spreading the light intensity of each ray to several adjacent samples instead of only one. This improvement is somewhat analogous to the one described by (4), with the exception that it must be applied to each pixel of projector image individually, instead of manipulating with already averaged luminance values. However, implementing this improvement may slow down the execution of the accumulative algorithm, and thus make it less suitable for real-time applications.

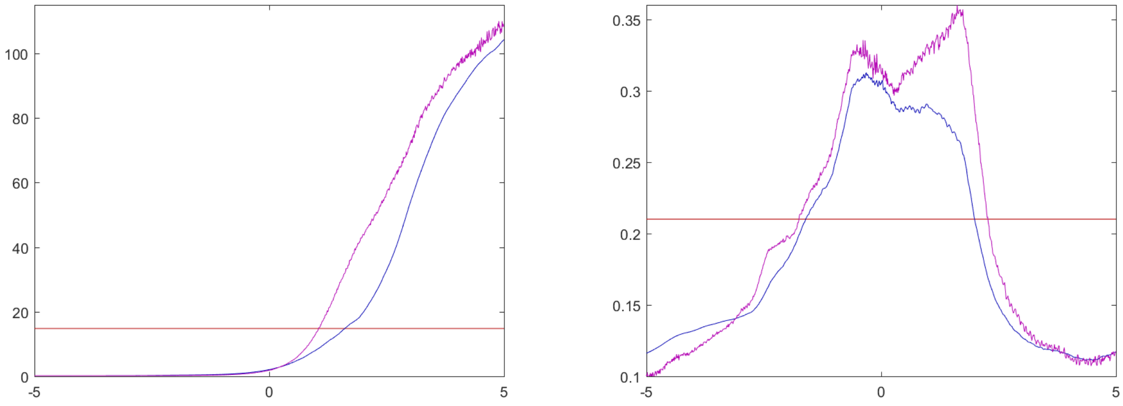
6. EXTENDING THE METHOD

Introduced approaches (brute-force and accumulative) can be extended to other types of 3D displays. Let us consider the cases of widely known LF displays, keeping in mind that the actual implementation may require additional adjustments.

6.1. Multi-view displays

Multi-view displays are usually parameterized with only few available observer positions/directions. Plus, the distribution of rays and their overall number is often very similar (or exactly the same) for all available views. In this case, it is not a problem to calculate the average global luminance for each view separately. If the calculation of the view index for a particular pixel is not trivial, one can create an additional single-channel integer-valued texture to store this index.

Also, it may be reasonable to calculate the average luminance using the mip-mapping. In this case, we need to create a two-channel square size texture for each individual view, and then apply method from Section 4.1.



(a) Scene “Sphere”.

(b) Scene “Pillars”.

Fig. 3: Estimated global luminance. Methods: All – red, BruteForce – blue, Accumulative – magenta. Horizontal axis shows the position on the observer line, vertical – estimated value.

Additionally, we can assume that each pixel in the input image affects only one pixel in the perceived image of particular view. In this case, we can initialize three-channel RGB texture as follows: channel R for view index, and channels GB for the pixel coordinates of perceived image. Using this texture, we can quickly initialize the set of perceived images, and apply mip-mapping approach from Section 4.1 to calculate the global average luminance.

6.2. Lenticular displays

Lenticular, as well as parallax-barrier displays, are often understood as the multi-view displays. In this case, speculations from Section 6.1 should be applied.

However, it is possible to parameterize the observer space not as a finite set of few discrete positions (like for multi-view displays). In this case, for each pixel of the input image, one can define the corresponding point in the observer space $X(\vec{r})$, similar to Section 3. After this, it is possible to apply the brute-force method from Section 4 to a particular points of the observer space. To apply the accumulative method from Section 5, we need to partition the observer space into non-overlapping regions, accumulate the luminance contribution for each partition, and calculate the average.

6.3. Full-parallax displays

All types of displays listed above may be both horizontal-only and full-parallax. In case of horizontal-only, the observer space is often represented as the observer line. This case was described in Sections 4 and 5. In case of full-parallax displays, the observer space may be parameterized with a plane.

With any kind of observer space parameterization, implementation of the brute-force method is the same: one need to pick an arbitrary point from the observer space, construct the estimated user-perceived image for this point, and calculate the average luminance for this image. To use the accumulative method, we need to partition the observer plane into non-overlapping regions, accumulate the luminance contribution for each partition, and calculate the average. Several types of partitioning may be applied, and the optimal one will depend on the actual display configuration.

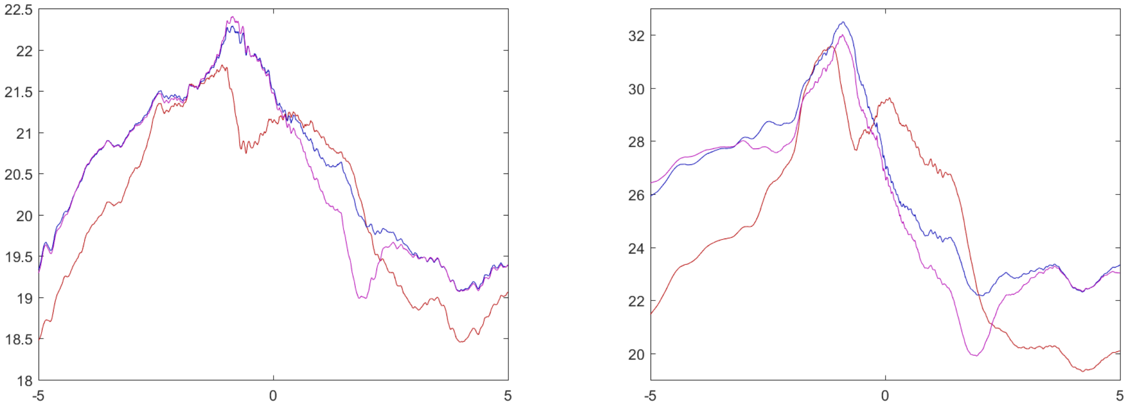
7. SIMULATION

7.1. Environment setup

We simulated the HoloVizio 80WLT full-angle display, with effective display size 645-by-360 mm, and expected distance to the observer line 2 meters. We took 10 meters range around the observer line center as the sampling area, and uniformly placed 1000 sampling points withing this range. Then, we applied separately the brute-force method and the accumulative method to estimate the global luminance level for each sample.

For the mentioned device model, we generated the set of projector images for two synthetic scenes: “Pillars” (see Figure 2) and “Sphere” (see Figure 5). Scene “Sphere” was constructed with the purpose of demonstration the drastic luminance change across the observer line. There are three point lights in this scene: right-most, with light intensity of 1000, and central and left-most with light intensity of 1. Scene “Pillars” was constructed to show “a typical” HDR synthetic scene. It has three directional lights illuminating the scene from behind, and self-illuminated HDR environment map.

We measured the global luminance method with different methods, for 1000 samples placed uniformly across the observer line, within 10 meters range around its center (see Fi-



(a) GroundTruth images.

(b) OnResult images.

Fig. 4: Values of the PSNR metric for the sequence of test image pairs, scene "Pillars". Methods: All – red, BruteForce – blue, Accumulative – magenta. Horizontal axis shows the position on the observer line, vertical – PSNR value.

gure 3). We tested the following methods of global luminance estimation:

- *All* – log-average luminance of combined 2D image of all projectors (see Section 4.1).
- *BruteForce* – brute-force method from Section 4.2.
- *Accumulative* – accumulative method from Section 5.1.

As it was pointed out in Section 5.3, the accumulative method has shown itself more sensitive to the oscillations of luminance level for small changes of the observer line position. Additionally, the accumulative method seems to give bigger amplitude of the estimated luminance than the brute-force.

7.2. Tone mapping setup

To show the impact of different global luminance estimation methods, we implemented the spatially-uniform version of Reinhard's tone mapping algorithm [8]. This method is based on per-pixel luminance adaptation, with final luminance level L_a being expressed as

$$L_a := L_s/(1 + L_s), \tag{5}$$

$$L_s := \alpha \cdot L/\overline{L}, \tag{6}$$

where L_s – scaled luminance, α – user-specified constant "key value", L – initial luminance, $\overline{L}$ – global average luminance.

In our case, global luminance level $\overline{L}$ may be different for each pixel in the projector images, since each pixel corresponds to its own position on the observer line. Once the value of $\overline{L}$ is estimated, we can apply formula (5) to adapt the luminance level for all projector images.

Additionally, we have generated the series of reference user-perceived tone mapped images:

- *GroundTruth* – actually captured HDR camera image of the synthetic scene, with subsequent application of Reinhard's tone mapping.
- *OnResult* – HDR camera images obtained through the simulation of LF display, with subsequent application of Reinhard's tone mapping.

7.3. Measurement

We have measured the PSNR metric for tone mapped image pairs across the observer line (see Figure 4), for test scene "Pillars". The first image in the pair is obtained by either GroundTruth or OnResult method. The second image – by applying (5) and (6) per each pixel of each projector image, with $\overline{L}$ being estimated by either All or BruteForce or Accumulative method. Figure 4 shows that BruteForce and Accumulative methods show similar performance most of the time. There are certain areas of the observer line, where PSNR for Accumulative method is noticeably worse than for BruteForce. Though, for most positions, Accumulative and BruteForce methods are both better than All method.

7.4. Visual comparison

Individual fragments of tone mapped images are shown on Figure 5 (scene "Sphere") and Figure 6 (scene "Pillars"). For scene "Pillars", it is clearly seen that the level of details for OnResult images is much worse than for GroundTruth images. The loss of detail naturally happens as the result of LF display simulation (it reflects the real-life behavior). This is especially noticeable for scene elements, that are supposed to be perceived at a large distance from the actual display surface. It can also be clearly seen that the scene details for All, BruteForce and Accumulative methods are rendered with higher quality than one would expect, if compared with OnResult images; but it is still worse than the GroundTruth images. Subjectively, we estimate the quality of images from BruteForce method to be superior to quality of All method, and Accumulative method to be even slightly better than BruteForce.

Similar observations are made for scene "Sphere" (Figure 5). For this scene, however, it is clearly seen that All method shows significantly worse result than BruteForce and Accumulative. Visually, BruteForce and Accumulative methods give almost identical results.

8. CONCLUSIONS

In this paper, we have shown that the problem of global luminance estimation is not trivial for the particular types of light field displays. We have introduced two methods to solve this problem (namely, brute-force and accumulative). Both methods work well for static synthetic scenes, and each has its own advantages and drawbacks. The accumulative method is light-weight, and can be executed in real-time. Moreover, this method can theoretically give identical result with the brute-force method after the proper adjustment.

Introduced methods can easily be extended to different types of light field displays. They can effectively be applied as a part of tone mapping algorithms.

9. ACKNOWLEDGEMENTS

The research in this paper was done as a part of and was funded from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 676401, ETN-FPI.

10. REFERENCES

[1] Tibor Balogh, Péter T. Kovács, and Zoltán Megyesi. HoloVizio 3D Display System. In *Proceedings of the First International Conference on Immersive Telecommunications*, ImmersCom '07, pages 19:1–19:5, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[2] Franklin C Crow. Summed-Area Tables for Texture Mapping. *ACM SIGGRAPH computer graphics*, 18(3):207–212, 1984.

[3] Gabriel Eilertsen, Rafał K. Mantiuk, and Jonas Unger. A comparative review of tone-mapping algorithms for high dynamic range video. In *Computer Graphics Forum*, volume 36, pages 565–592. Wiley Online Library, 2017.

[4] Gabriel Eilertsen, Robert Wanat, Rafał K. Mantiuk, and Jonas Unger. Evaluation of tone mapping operators for hdr-video. In *Computer Graphics Forum*, volume 32, pages 275–284. Wiley Online Library, 2013.

[5] Søren Forchhammer and Mantel Claire. Viewpoint adaptive display of HDR images. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1177–1181. IEEE, 2017.

[6] Zicong Mai, Colin Doutre, Panos Nasiopoulos, and Rabab Kreidieh Ward. Rendering 3-D high dynamic range images: Subjective evaluation of tone-mapping methods and preferred 3-D image attributes. *IEEE Journal of Selected Topics in Signal Processing*, 6(5):597–610, 2012.

[7] Hossein Najaf-Zadeh, Madhukar Budagavi, and Esmaeil Faramarzi. VR+HDR: A system for view-dependent rendering of HDR video in virtual reality. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 1032–1036. IEEE, 2017.

[8] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. *ACM transactions on graphics (TOG)*, 21(3):267–276, 2002.

[9] Marcos Slomp, Michihiro Mikamo, and Kazufumi Kaneda. Fast Local Tone Mapping, Summed-Area Tables and Mesopic Vision Simulation. In *Computer Graphics*. InTech, 2012.

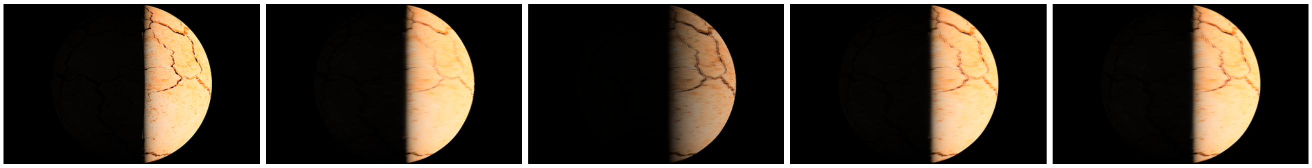
[10] Marcos Slomp and Manuel M. Oliveira. Real-Time Photographic Local Tone Reproduction Using Summed-Area Tables. In *Computer Graphics International*, pages 82–91, 2008.

[11] Cees van Berkel. Image preparation for 3D LCD. In *Stereoscopic Displays and Virtual Reality Systems VI*, volume 3639, pages 84–92. International Society for Optics and Photonics, 1999.

[12] Peng Wang, Xinzhu Sang, Yanhong Zhu, Songlin Xie, Duo Chen, Nan Guo, and Chongxiu Yu. Image quality improvement of multi-projection 3d display through tone mapping based optimization. *Optics express*, 25(17):20894–20910, 2017.

[13] Masahiro Yamaguchi. Light-field and holographic three-dimensional displays. *Journal of the Optical Society of America (JOSA)*, 33(12):2348–2364, 2016.

(a) GroundTruth, left. (b) OnResult, left. (c) All, left. (d) BruteForce, left. (e) Accumulative, left.



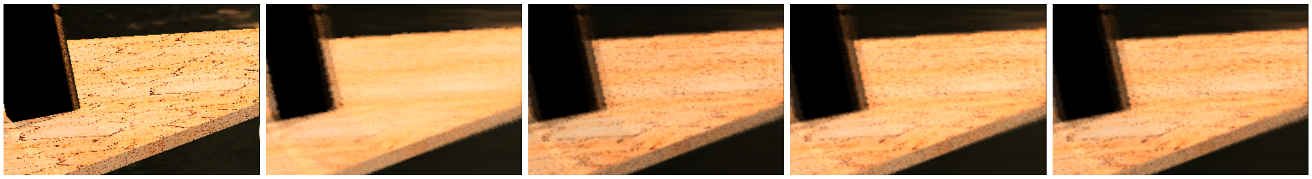
(f) GroundTruth, center. (g) OnResult, center. (h) All, center. (i) BruteForce, center. (j) Accumulative, center.

(k) GroundTruth, right. (l) OnResult, right. (m) All, right. (n) BruteForce, right. (o) Accumulative, right.

Fig. 5: Pinhole-camera images after Reinhard's tone mapping, scene "Sphere".

(a) GroundTruth 1. (b) OnResult 1. (c) All 1. (d) BruteForce 1. (e) Accumulative 1.

(f) GroundTruth 2. (g) OnResult 2. (h) All 2. (i) BruteForce 2. (j) Accumulative 2.



(k) GroundTruth 3. (l) OnResult 3. (m) All 3. (n) BruteForce 3. (o) Accumulative 3.

Fig. 6: Fragments of pinhole-camera images after Reinhard's tone mapping, scene "Pillars".