To:

Editor in Chief, Computer Graphics Forum

Associate Reviewers

Dear Editors, dear Reviewers,

This letter accompanies the minor revision of our paper “Hybrid Data Visualization Based on Depth Complexity Histogram Analysis”, CGF-14-EG-016.R1, (following a fast-track major revision from EG2014 submission). Below, we provide an account for how we addressed the reviewers’ comments to our initial submission (review excerpts are set in blue italics). The associated changes are also highlighted in green in the revised manuscript (pdf file).

Reviewer 1

(References) Adding one or two sentences describing the original a-buffer method could improve readability further.

> The introduction of the A-buffer concept in Section 2 was revised with the following addition: “The A-buffer is a leading OIT solution based on the temporary storage and sorting of intermediate pixel fragments”.

(Technical Soundness) Section 3.1 should be improved to better understand how a DCH is computed, right now this is rather unclear and confusing.

> Section 3.1 has been reworked and now utilizes the protein in Figure 1 to clarify and exemplify the process of creating a DCH. The presentation is now also more chronological in that it first describes the creation of the intermediate complexity image before describing how this leads to a DCH.

(Exposition, Figures) The abstract and introduction are very vague in regards to the paper's contribution. No details at all are given about the 2 proposed methods. It is stated several times that 'two novel components' will be introduced, but some more information on what that means would be good.

> Appropriate parts of both the abstract and the introduction were revised such that the two components now are discussed separately rather than collectively. Additional details were also provided.

Reviewer 2

"current state-of-the-art algorithms do not support interactive exploration as complexity increases" I would argue that this statement is a little bit too strong. After all the proposed method does not speed up the state-of-the-art by orders of magnitude.

> The statement has been rephrased, now acknowledging that state-of-the-art techniques may be sufficient even if this is not always guaranteed.

"2. Related Work", it may sound a little far fetched and is really just a recommendation but you may to consider adding a reference to a modern "ray-guided-VR" paper (such as Hadwiger et al.

"Interactive volume exploration of petascale microscopy data streams using a visualization-driven virtual memory approach") mainly because multi-volume rendering with such an approach is a trivial extension of a few lines of shader code.

> While Hadwiger's work is very impressive, we did not find any significant connection between their work and ours other than that both employ volume rendering. Extending Hadwiger's work to multiple volumes may be possible, but we are not ready to make any claim that our code could facilitate this without such a claim being validated. Given the size of Hadwiger's framework, such a validation would require prohibitively extensive implementations given its marginal significance to our paper.

As for depth peeling, you may want to consider adding a few more recent references to you list of papers to demonstrate that this is a very active field of research and not something that none has worked on for half a decade.

> The following references were added to strengthen the case that both A-buffers and Depth Peeling are very much active areas of research: Kerzner2013, Vasilakis2013, Yu2013, Vasilakis2014.

"3.1. Depth Complexity Histograms" I'm just curious: can't you just use the accumulation buffer for that purpose? Is that still supported by todays hardware?

> We do not believe the accumulation buffer is applicable here as the computation of the depth complexity image relies on geometry-to-buffer operations rather than buffer-to-buffer operations. That said, integer buffers and atomic operations are becoming core GL functionality and should be available on a wide selection of hardware in the near future.

"5.2. Preventing Over-sized Local Arrays Using per-pixel Depth Peeling" is the problem of overflowing depth peeling buffers really such a big issue? How big is the difference if only the first n (say 32) surfaces are rendered?

> We would argue that this is a question which need to be asked on a per-case basis. There are naturally cases and settings where the first 32 (or even 8) layers comprise the majority of the pixels' final color. But unless we can guarantee that this is always the case, then completely removing the dependency of an upper boundary is still a step forward. We believe it is significantly better that an upper boundary is an optional performance trade-off rather than an enforced necessity. Also, the overflow prevention is only one of the benefits as this particular component, as it also provides increased performance for 32-and-lower sized arrays.

In addition, spelling changes and other minor reformulations are highlighted in yellow in the revised manuscript (pdf file). This includes a slight shortening of the conclusions section due to page overflow after addressing the reviewers' comments.

We are grateful for the detailed reviews and we hope that we were able to address all concerns adequately.

Yours sincerely,

The Authors