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” (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).

**Reviewer 1**

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

The introduction of the A-buffer concept in Section 2 was revised.

(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 more clearly references to the corresponding sub figures of Figure 1 with the aim to exemplify the process of DCH creation.

(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.

Both the abstract and the introduction were revised such that the two components now are discussed in turn rather than collectively. A bit more detail is 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 sentence has been rephrased and now acknowledges that there may be cases where state-of-the-art techniques are 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 et al's work is very impressive, we did not find any stronger connection between their work and ours than that both include volume rendering. Extending Hadwiger's work to multiple volumes may be possible, but we are not ready to make such a claim without validation, which, given the size of Hadwiger's framework, would require prohibitively extensive implementations given the marginal connection to our contributions.

*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.*

Kerzner2013, Vasilakis2013, Vasilakis2014, Yu2013

*"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 requires per-fragment atomics.

*"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?*

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 the pixels are 100% opaque for all scenes and settings before a certain depth level then completely removing the dependency of an upper boundary is still a step forward. We believe it is always better that an upper boundary is an optional performance trade-off than an enforced necessity. Also, the overflow prevention is only one of the benefits as the component also speeds up the computation for 32-and-lower sized arrays.

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

Yours sincerely,

The Authors