**COVER LETTER**

**Submission ID**: 1519

**Title**: Accelerated Visualization of Transparent Molecular Surfaces in Molecular Dynamics

**Date**: January 4th, 2016

We would like to thank the anonymous reviewers for their helpful comments. We attempted to address all remarks in the revised version of the paper. In this document, we give detailed responses to all reviewer comments and describe the corresponding changes in our manuscript.

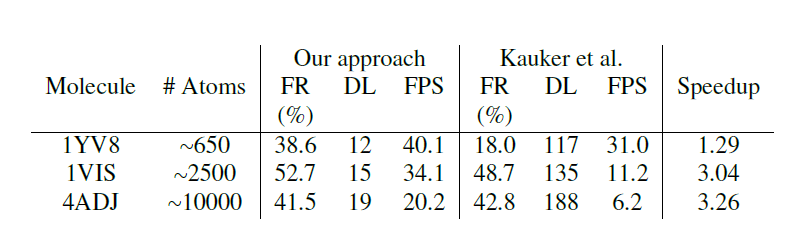
**Summary of all reviews from the Primary Reviewer**

1. The authors should explain how they chose this value (24 depth layers) and how the number of depth layers affects the performance. As for the 24 depth layers, I also think that this is no big deal, but there is still the issue of correctness. Of course the authors could argue that 24 depth layers are enough because more depth layers are not distinguishable anymore, but my point was that the authors do not comment on this at all. For smaller molecules (like the ones shown in the paper), probably even less depth layers could be used, which could improve the performance. Therefore, a couple of measurements how many depth layers are actually needed and how changing this number affects the performance would be interesting. However, as I already wrote, I also think that this is not a central point here.

The value (24 depth layers) was only the limit of our simplistic implementation. The per-pixel linked lists of fragments (as described in Yang et al., 2010) limit the total number of stored fragments (by available memory size), but not the maximum length (depth) of individual lists (pixel). Therefore, we decided to completely remove the sentence from the text.

1. Comparing the performance with [12], [13] is okay for the discussion but not fair for the real results. Rendering the SES opaque or with simple blending is much easier than using full transparency. If I see this correct the performance of Kauker is the only comparable. For 1VIS we have 10.2 vs. 16.4 fps, where the 16.4 includes not only the ray casting but also the surface computation. If in the work of Kauker the pure ray casting is measured we even have 10.2 vs 25.4 on the same graphics card, which is more than a factor of 2.

We completely agree with this comment. Our new results even show that our speedup is a factor of 3, which is visible from Table 2:



**Primary Reviewer**

1. It is worth to briefly discuss transparent visualization as one option among others to show the outer surface and reveal the inner surface at the same time. The papers from Alethea Bair may serve as orientation. Illustrative techniques, used by Tarini:2006 and Lawonn:2014 are an alternative. Besides this suggestion, I have the  
   impression that some more recent papers should be briefly discussed. Here  
   is what I found when looking for further related work.

All suggested papers, except for [Lindow et al., 2014], were added to Chapter 2 Related Work. Lindow et al. describe LES which yet is a novel surface representation, but on the other hand it is beyond the scope of this paper, since we address only the SES representations.

1. I would explain the Fourier window.

The Fourier window was mention in the related work where we tried to briefly explain the A-buffer principle. However, we agree that this would need more space to explain it properly. Therefore, we decided to remove this notion about the Fourier window and use the space for other citations requested by the reviewers.

1. Minor comments: At page 5: the values of f -> f\_i.   Page 5: user defined -> user-defined.

Corrected.

1. I would emphasize the discussions of limitation with a paragraph with that name.

We added such a paragraph to the Performance Analysis chapter, as the performance is one of our main benefits. As we also changed the implementation, many previous limitations were removed so the text changed substantially as well. Other limitations of our approach, from the biochemical point of view, are also mentioned in section 6.2 User Feedback. However, these are related to the applicability of our algorithm to different tasks, which was not the key goal of this paper.

**Review 2**

1. As mentioned above, the authors heavily rely on previous work. Therefore, I think their claim that they "presented a new approach for the real-time computation and visualization of the transparent molecular surface and the exact representation of inner cavities" does not hold. Cavity extraction using the SES is not "a novel [...] algorithm for cavity extraction" (see, e.g., [MR1, MR3]). In fact, even the popular MSMS software [MR2] extracts inner voids defined by the SES. Therefore, I would rate the contribution and originality of the paper as quite low.

We agree with the reviewer that this claim, presented in the Conclusion section (as well as in the Introduction) is incorrect. We did not propose a novel algorithm for extraction of cavities, our main contribution lies in the introduction of an accelerated rendering pipeline for the computation and visualization of transparent SES. Therefore, this was corrected in the Introduction as well as in Conclusion and the suggested missing references were added.

1. The paper also does not present a significant improvement in terms of computation speed compared to previous work [12, 13]. In fact, the rendering times for the ray casting are much lower than the ones reported on previous works. This is probably due to the more elaborate intersection tests that are required to remove interior parts of the spheres and tori. I think that using tighter bounds for ray-casting (as suggested by the authors) would probably not improve the performance significantly.

It might not been clear from the paper, that our contribution does not lie in providing an accelerated technique for SES computation. Instead, we significantly extend the existing technique [Krone et al., 2011] to enable transparent SES rendering. Our transparent rendering pipeline is accelerated compared to [Kauker et al., 2013].

In Section 6.2, there was the sentence:

“We also utilize OpenCL to accelerate a kernel which computes the positions of spherical triangles in the original algorithm [Krone et al., 2011]”.

The sentence may be misinterpreted in a way that it states that our implementation of the mentioned part of the algorithm was accelerated compared to [Krone et al., 2011]. Instead, it should only state that we implemented that specific part using OpenCL instead of GLSL, because its GLSL implementation was significantly slower than the original one by Krone et al. In fact, the OpenCL implementation of this part differs only very little in the computation speed compared to the original algorithm by Krone et al.

Therefore, we changed the sentence in order to make this clear:

“We also utilized OpenCL to implement a kernel which computes the positions of spherical triangles in the original algorithm [Krone et al., 2011]”.

We also included Table 1 that shows that our assumptions about the possible speedup by implementing tighter bounding boxes for ray-casting are correct.

1. In my opinion, the authors should focus more on the data structures and computations for the transparent rendering (Section 4.1), since this is the only real contribution of the paper.

We focused more on the implementation description when we considered it to be necessary (see our answer to comment no. 10).

1. Minor comment: Many technical/implementation details are missing, in particular in Sections 4.1 and 5.1, which are, in my opinion, the most interesting sections of the manuscript. For example, it is not entirely clear to me how the spherical patches are ray-cast.

The implementation details were extended significantly by the information about the speedup of ray-casting.

1. Minor comment: The authors mention that they employ the bubble sort algorithm for sorting, which has much quadratic runtime complexity. There are faster sorting algorithms, which have the additional benefit that they are parallelizable. It would be interesting to hear why the authors opted for the subpar bubble sort. Are the data so small that it does not matter?

This is exactly the case, we are dealing with such a small data that the selection of the sorting algorithm does not influence the performance at all. We apply the sorting for molecules consisting of 10,000 atoms, where the maximum number of items to sort is 8.

1. Minor comment:   The visibility sphere for toroidal patches (page 5, first paragraph & Fig. 7) was introduced by Krone et al. [11]. The authors should mention this and cite the paper, since it gives the equations to compute this sphere.

We agree that the citation was missing. So we included the following sentence to Section 4.2.1:

“The clipping of fragments in the affected spherical patches can be done using a clipping plane or a visibility sphere (introduced in [Krone et al., 2009]) of a torus.”

1. Minor comment:  The case study (Section 6.1) does not offer any new insights. I think this could be shortened to the user feedback at the end of the section. Furthermore, I do not see why a user should chose a clipping plane when creating an offline animation, since molecular visualization packages like VMD can also render semi-transparent surfaces (using ray tracing).

We agree that the presented case study did not bring any breakthrough and rather presented one of the possible usages of our approach. Therefore, we decided to remove it. Now Chapter 6 contains a brief discussion about the results, followed by the performance measurements and user feedback (which was slightly extended).

1. Minor comment:  The authors use an approximate area calculation (Sec. 5.3). The author’s should comment on this choice. Connolly [4] presented the equations to compute the exact surface area of the SES, so there seems to be no need to resort to estimations.

We use the area calculation only for coloring the extracted cavities and their further filtering. For this purpose the approximated value is sufficient. As our algorithm primarily does no focus on the cavity computation and extraction and presentation of their parameters, we did not consider a better solution. But we completely agree with the reviewer that the exact values are very useful for the domain experts and this is very good suggestion for the further improvement of our solution. Therefore, after discussing it with the biochemists, we added the notion about it to the User Feedback section.

1. Minor comment: There seem to be bugs in the renderings shown in Fig. 9: in the leftmost image, an orange part is clearly visible, while the outer surface is supposed to be turquoise. In all of the images, single-pixel artifacts are visible. The authors should comment on this. Is this a problem with the floating-point precision? Do rays miss the patches? These artifacts are also visible in Fig. 1 & 10.

In Figure 9 there was a bug where the protein surface was clipped by the surface of the cavity. We fixed this, replaced the image, and commented it in the paper (Section 5.2). We also commented on the single-pixel artifacts (visible in Figures 1 and 10) as the reviewer suggested (Section 5.1). However, Figure 10 was removed from the paper according to the suggestion of reviewer no. 4.

**Review 3**

1. Even though the authors assert that their system is the first that enables interactive visualization of this particular data, the building blocks of their system are standard computer graphics techniques. I didn't find much new methodology in this paper from which others could learn for building their own systems.

We agree that this was not clear in our description (which was also stated by reviewer no. 4). Therefore we updated the description of our main contributions in the Introduction section. Now we state more clearly that the novelty of our approach lies in the introduction of an accelerated rendering pipeline for the transparent SES which then enables to explore the inner cavities in molecular dynamics in real-time.

1. It wasn't obvious to me whether it is indeed highly important to the intended users' day-to-day work to be able to create the final visualizations fully interactively. Some simplified interactive "mockup" that would allow them to interactively adjust the viewpoint and clipping planes (without providing the full detail of the final rendering, which would still be done off-line) might have worked similarly well for them, and it might have been possible to create it with very little effort.

We agree that the fully interactive real-time visualization is not the most critical part of the day-to-day work of biochemists. By introducing the case study we aimed to demonstrate a possible usage of our approach and it obviously led to misinterpretation. Thus, also according to the suggestion of reviewer no. 4, we decided to remove the case study from the paper. As for the suggested mockups, this is actually a very interesting idea. But as we are not aware of any available solution for this domain, we decided not to mention it in the paper.

1. What was your motivation for using bubble sort, which is known to scale poorly to larger numbers of items?

As already mentioned in the answer to comment no. 11, this sorting algorithm was selected because we are dealing with only very small datasets (containing maximum 8 items).

1. Why does enabling semi-transparency increase the framerate (according to the caption of Fig. 1)? Should framerates be specified with standard errors?

Here we apologize but we made a mistake in the measuring phase. The framerate for the opaque surface was measured also with the inner cavities (which were invisible, but still present and taking a significant portion of performance). Therefore, we corrected this mistake.

1. Please rephrase: p.6: "The dynamic movements of the molecule can cause its rotation and the clip plane settings in the first time step might have completely wrong positions in the following steps."

This sentence was completely removed because, according to the suggestion of reviewer no. 4, we completely removed the case study as it was considered to be redundant. We replaced it by more thorough description of the algorithm, new related work and extension of the possible future directions.

1. Typos: "cannot spent" -> "cannot spend" "by ray-cast" -> "by ray-casting" "Another possible extensions" -> "Other possible extensions"

Corrected.

**Review 4**

1. First, the authors should be more careful with the description of their contributions. The described approach is a combination of existing techniques, where certain steps are improved or accelerated. I would not rate the cavity detection that high because it does only color parts of the surface that belong to closed cavities of the molecule. On the one hand visualizing only these parts is not sufficient for a good visual cavity analysis. One can see this in the video. Cavities can appear and disappear form one time step to the other or when changing the probe. But what really happens is that in most cases the cavity becomes accessible from outside the molecule (like a channel, tunnel, or pocket). Often these cavities are even more interested than internal closed cavities and the appearance and disappearance is sometimes confusing and disturbing. On the other hand one cannot really analyze the cavities in detail, only the area is roughly estimated. Just to clarify, I think the analysis of closed cavities is a good feature, but in the context of molecular cavity analysis, it cannot compete against some of the other approaches out there. In my opinion, the main contribution is the accelerated and improved rendering pipeline for the transparent SES.

We completely agree with the reviewer, the list of contributions was not correct. Therefore, we corrected this information in the Introduction section:

“The main contribution of our solution consists of the introduction of the accelerated rendering pipeline improving the performance of visualization of transparent solvent-excluded surface (SES) of the molecule (in comparison with Kauker et al. [13]). To provide a better perception of inner surface features (cavities), we allow users to change the parameters of the opacity modulation. Moreover, we extended the existing state-of-the-art approach [16] to SES computation by proposing methods for computation and rendering of individual SES patches.”

1. In my eyes the related work for molecular surface renderings is sufficient. The cavity detection part requires some revision. I expected more approaches that deal with molecular dynamic trajectories. For example the works by Krone et al. that can also handle dynamic data in real-time but do not have the restriction to closed cavities. Or the works by Lindow et al. that can trace cavities in real-time with more geometrical accuracy than the approaches by Krone et al. but require some pre-computation. Additionally I expected the approach by Borland et al., where ambient occlusion was used to set-up the transparency of the surface.

We added the required references and discussed their benefits.

1. The authors should consistently use either the term surface patch or surface primitive. I think the term "patch" is much better because primitive is often used in computer graphic for simple geometric structures (points, lines, polygons). Here we have polynomial patches. Furthermore, the authors should clarify in one or two sentences the definition of a patch. Typically, a patch of the SES is one closed surface component. Thus, some convex spherical patches belong to the same atom sphere and some toroidal patches belong to the same torus. However, if I understand the authors correct, they define, at least sometimes, a toroidal patch as the set of all toroidal patches that connect the same two atoms and a convex spherical patch as all convex patches that belong to the same atom. Maybe the authors can add a clear mathematical description of the surface and its patches before starting the description of the method.

We agree with the reviewer that we did not use these terms carefully. Therefore, we revised the paper and carefully corrected it.

We also agree that a clear mathematical description would be appropriate but due to the lack of space we decided not to include this into the paper. The description can be found in mentioned related work [Connoly, 1983].

1. During the description of the surface computation, the authors write that they introduce a "novel" data structure which is in the end a hash table. Is a hash table really novel? The authors should directly write the purpose for the hash table (better memory efficiency than others) instead of the "novel data structure". This makes everything more clear.

We completely agree, this was incorrect. What we wanted to say was that the data structure we are using is novel with respect to that one used by [Krone et al., 2011]. With our hash table, we reach memory savings. The information was corrected in the paper.

1. The surface graph is the graph of the contours of the SAS. The authors should maybe mention it in this way. Furthermore, they write that an adjacency matrix is used. Typically a classical adjacency matrix is a quadratic matrix with a column and a row for each vertex and an entry unequal to zero if two vertices are connected. In contrast the authors describe a classical adjacency list, but since every vertex has 3 edges one can store it in a matrix with three columns. This can be confusing and should be clarified. Overall the definition could be a bit more formal here. Are the entries integer indices for the array of edges E? It was not completely clear to me.

We agree with the reviewer that the description of our data structure which we called the surface graph is clearer when the term graph of SAS contours is used. Therefore, in the paper, we state that the surface graph is the analogy to the graph of contours that defines the SES patches (Section 4.2).

We also agree with the reviewer that we interchanged the term adjacency matrix with the adjacency list. We now use the correct term adjacency list in the description of the cavity computation algorithm (Section 4.2).

Finally, we tried to make the description of the adjacency list implementation clearer. We added that the entries in the matrix are indices to the array of vertices V (Section 4.2).

1. Figure 2 does not really reflect the visualization pipeline as described by the authors. Stage 1 is the data, stage 2 the surface computation, stage 3 the separation into outer surface and closed cavities, stage 4 (only for the cavities) is the area estimation and filtering, stage 5 is the ray casting of all patches, and stage 6 is the transparency visualization based on the fragment linked list. Either I misunderstood something or the authors should revise the figure. Additionally I would suggest to use the same molecule for all depictions in this figure, otherwise it can be confusing.

Thanks to this comment, we discussed a lot the content of Figure 2. Our aim was to follow the pipeline in the image as well as in the text. We agreed that the image was unclear so we decided to simplify it and we also added links to corresponding chapters and sections to the figure caption. The image covers the suggested stages, however, we decided not to highlight all six stages because of the consistency with the text.

1. The algorithm to check if a point lies inside a convex spherical patch seems to be too complicated. The test with the even-odd rule requires a lot of computation. But it is only necessary to test if the point lies inside the convex polyhedron given by the planes in which the contour arcs lie. In more detail: each convex spherical patch is just a scaled version of the corresponding SAS patch. It was described by Varshney et al. that the power cells of the SAS-atoms bound these patches. In other words, the contour arcs of the SAS lie in the faces of the power diagram of the SAS. Each power cell is a convex polyhedron. Scaling the polyhedron gives the boundary for the corresponding SES patch. Since it is convex, it is just a check if a point lies inside a convex polyhedron. I hope my description is clear enough. I know that people used this polyhedron-based test successfully during triangulation of the SES.

This comment pointed us to the literature which covers the triangulation of the SES. We would like to have a specific reference to such a paper as we were not able to find a paper which describes the problem we are dealing with. Nevertheless, we decided to replace the point in polygon (PIP) check with the method suggested by the reviewer. We did that for two reasons:

* The PIP check was a source of single-pixel artifacts in the final image as we did not handle the case when the ray hits a polygon vertex.
* From our experiments it seems that two (or more) spherical patches belonging to the same atom sphere can be bound using convex bounds (obtained by cutting the polyhedron of sphere’s power cell) that do not intersect themselves. Then, a spherical patch can be ray-cast as a sphere cut by the planes of its convex polyhedron bound.

1. During the area estimation not only the toroidal and convex spherical patches are ignored but also the singularities. The estimation is still okay for visualization but the authors should mention this. Overall they should describe in a few sentences the singularity handling.

We are aware of this but we are ignoring the singularity handling when computing the area estimation. The reason is that we are using the area estimation only for coloring of cavities (see also answer to comment no. 14).

1. I suggest to separate discussion and results (starting with results). Additionally, I expected some comparisons with the works by Kauker et al. and Krone et al.

Chapter 6 was reorganized so now it starts with the description of our main results followed by the description of performance and discussion summarizing the opinions of our cooperation partners from computation biochemistry.

The comparison was added to Table 2, along with its description.

1. I also suggest to remove the case study and Fig. 10. The space can be used for further figures and more detailed descriptions, better related work and more comparative results. It would be much better to have more details about the implementation such that the approach can be easily reproduced, which is currently difficult. The case study is not really state of the art. Our cooperation partners create videos, similar as described in the study, in just some hours or even less time. And although older techniques are not able to handle transparent visualization in real-time, one can check several time steps over the trajectory in a few minutes such that there is also no trial and error phase. Or they use other real-time techniques, like the secondary structure representation or the ball-and-stick representation. Furthermore, I would not claim that transparency is in general better than clipping. Both techniques have their advantages and drawbacks. I made the experience that our cooperation partners often prefer clipping planes. Transparency rendering can be confusing in certain cases. This is a methods paper which clearly improves the current SES rendering. A case study like described here is not necessary.

We completely agree with this so the case study, including Fig. 10, was removed from the paper. We also included a notion about the traditionally used technique of checking few time steps to make an overview and how our technique can improve this process.

We also agree that our claim that transparency is better than clip planes was misleading. Definitely there are cases when the clip planes are more advantageous. So we removed this claim and replaced it by one case where the surface is substantial – studying interactions between proteins.

1. Maybe the title can be more concrete.

We changed the title and we believe that now it better reflects the main goal of our paper. The new title is “Accelerated Visualization of Transparent Molecular Surfaces in Molecular Dynamics”.