**COVER LETTER**

**Submission ID**: 1519

**Title**: Real-Time Visualization of Protein Surface Features 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.
2. 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.

**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.
2. I would explain the Fourier window.
3. 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.

**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.
2. 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.
3. 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.
4. 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.
5. 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?
6. 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.
7. 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).
8. 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.
9. 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.

**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.
2. 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.
3. What was your motivation for using bubble sort, which is known to scale poorly to larger numbers of items?
4. Why does enabling semi-transparency increase the framerate (according to the caption of Fig. 1)? Should framerates be specified with standard errors?
5. 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."
6. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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 trail 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.
11. Maybe the title can be more concrete.