EuroRV3: EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization (2015)
W. Aigner, P. Rosenthal, and C. Scheidegger (Editors)

On the Reproducibility of *Line Integral Convolution for Real-Time Illustration of Molecular Surface Shape and Salient Regions*

Kai Lawonn[1,2], Michael Krone[3], Thomas Ertl[3], and Bernhard Preim[2]

[1]Computer Graphics and Visualization, TU Delft, The Netherlands
[2]Department of Simulation and Graphics, University of Magdeburg, Germany
[3]Visualization Research Center, University of Stuttgart, Germany

Abstract

In this paper, we discuss the reproducibility of our work presented at EuroVis 2014 [LKEP14], which describes an illustrative rendering method tailored to molecular surfaces. We distinguish between the reproducibility of the data sets that were used for figures and performance analysis and the reproducibility in the sense of re-implementing the method. For the latter, we focus on each step of the algorithm and discuss the implementation challenges. We give further details and explain the most difficult parts. Additionally, we discuss how the models that were used can be recreated and the availability of the underlying data. Finally, we discuss the current state of reproducibility of our method and reflect on the problem of offering the source code of a research project in general. The definitive version is available at `http://diglib.eg.org/`.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

At EuroVis 2014, we presented the illustrative visualization technique shown in Figure 1, which conveys the shape of molecular surfaces in real-time [LKEP14]. We used line drawing techniques that can be applied on animated surfaces in a frame-coherent manner. The technique was designed to highlight the structure of molecular surfaces and to illustrate important surface features like cavities, channels, and pockets. The core of our method consists of three main steps:

1. Feature vector field: We compute an illumination-based vector field.
2. Feature region: We determine a scalar field that represents salient regions based on the feature vector field. Optionally, a second feature scalar field is determined.
3. Line integral convolution: We adapt the LIC to the salient regions along the vector field of step 1.

In the following, we review the reproducibility of our method and discuss possible improvements. First, we discuss what kind of data we used and where it can be found.

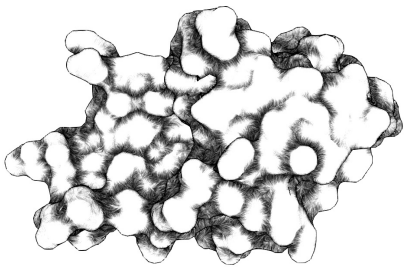

Figure 1: *Our illustrative rendering method [LKEP14] applied to the molecular surface (QuickSurf [KSES12], default parameters) of an isomerase (PDB ID: 1OGZ).*

2. Molecular Data Sets

Most data sets that were used in our paper are based on freely available protein data that can be downloaded from the Protein Data Bank (PDB) [BWF*00]. In addition to the scientific name of the protein, each protein structure in the PDB has a unique four-character identifier (the so-called PDB ID). This ID is important since there are often many different data sets that contain a protein with the same name. These data sets can for example be different conformations of the

same protein or proteins from different species that have the same function. In the first case, only the spatial arrangement of the atoms differs. In the second case, the atom composition that forms the protein can also differ. We only mention the name of the protein, which makes it hard to find exactly the same data set. In some cases, the molecular structures used in our figures and measurements originate from molecular simulations that were run by our project partners. This data is not freely available to the public, which is a further limitation. However, since proteins with the same name usually differ only slightly, one can use the name of the protein to find sufficiently similar data sets in the PDB. Therefore, we would argue that the reproducibility is still reasonable.

As mentioned in our paper [LKEP14], the molecular surface meshes were generated using the freely available molecular visualization tool VMD [HDS96]. In particular, we used the van der Waals surface (drawing mode *VDW*), the Solvent Excluded Surface (drawing mode *MSMS* [SOS96]), and *QuickSurf* [KSES12], as also stated in our paper. Using a freely available tool for data generation and referencing the drawing modes makes it easy to reproduce the data. However, the table that shows the measurements for the data only gives number of triangles for each mesh, which requires to "reverse engineer" the parameters. In hindsight, it would be beneficial to also give the exact parameters that were used.

3. Methods

In the following, we describe the single stages of the algorithm and discuss their reproducibility.

Feature Vector Field The first step of our algorithm is to determine a feature vector field. For this, we used known algorithms from discrete differential geometry. The goal was to determine a gradient for each vertex based on the diffuse illumination. We explained the underlying formula and cited two references. The only difficulty might be how to transfer it to the GPU. For this, we used a data structure that was also cited in the paper [LKEP14] that uses *shader storage buffer*. The calculation is easy to follow, the only drawback is to re-implement this stage. We guess that only an experienced developer is able to implement this approach on the GPU.

Feature Regions This step is based on the feature vector field. Here, we need the covariant derivative of the diffuse illumination along the projected view vector. As this calculation is based on suggestive contours [DFRS03], more details can be found in the original work. For this curvatures are needed, but as we wanted to apply that calculation on animated surfaces, curvature calculation is not an option on the GPU. Fortunately, they used an equivalent equation, which is based on the lighting only. If the vector field is determined, this calculation is much easier.

The second (optional) scalar field we used encodes screen space ambient occlusion (SSAO). As mentioned in the paper, we opted for the SSAO method by Kajalin [Kaj09]. The original publication describes the implementation in detail and even includes the pixel shader source code. Therefore, we would rate the SSAO as easily reproducible.

Line Integral Convolution The line integral convolution (LIC) algorithm is based on the work of Huang et al. [HPW*12]. We recapitulated their algorithm to point out some different parts of the noise texture generation. This step is highly reproducible as it is simply a recursive algorithm to generate the textures. Furthermore, a figure in our paper shows the first eight textures, which improves the reproducibility. As defined in the original work, in the subsequent texture projection, each triangle is projected onto the noise texture. For this, the texture is uploaded to the GPU and the corresponding pixels are used for the triangles. The most difficult part was the propagation of the LIC, which requires to project the vector field to the view plane. We store the projected illumination gradient values in textures for the front and back faces using frame buffer objects (FBO). The subsequent computation of the LIC is straightforward. Using the FBOs allows us to aquire the LIC in screen space.

4. Discussion

Overall we presented our algorithm in a clear way such that an experienced master student should be able to implement the method. As for all GPU-based algorithms, a basic knowledge of GPU programming is required. Our paper includes a schematic overview of the main steps of our algorithm and we provided the noise textures for further use. The main steps of our algorithm are based on other papers that are cited, which helps to implement these steps. We did, however, not present or offer the source code of our project. The main reason for this was that scientific work mostly aims at conferences, which have a deadline. Therefore, there is often no time to write clean, commented source code and a documentation. Another problem is that our work is written as a module for *MevisLab* [mev]. To use our module, the user has to download the full *MevisLab* framework, which is about 1 GB big. Afterwards, the user has to compile our code in order to use the module in the software package. We consider to use small, standalone applications in the future since they make it easier to share research prototypes as open source projects. On the other hand, using an existing visualization framework like *Mevislab* that offers a lot of basic functionality leads to lower development times, which is beneficial for the research process. In general, providing selected source code fragments or pseudo-code would be a good tradeoff to improve the reproducibility, but the code listings would take a lot of the (limited) page space of a paper.

We presented the result of our performance measurements in a table, which includes the number of triangles and the frames per second. Therefore, the reader is able to assess and reproduce the general performance results of our technique.

In summary, we would rate the implementation of our method as well reproducible based on the details provided in our paper and in the cited references. For a faithful reproduction of the results, the description of the data sets should be more detailed.

References

[BWF*00] BERMAN H. M., WESTBROOK J., FENG Z., GILLILAND G., BHAT T. N., WEISSIG H., SHINDYALOV I. N., BOURNE P. E.: The Protein Data Bank. *Nucleic Acids Research 28*, 1 (2000), 235–242. URL: http://www.pdb.org, doi:10.1093/nar/28.1.235. 1

[DFRS03] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S., SANTELLA A.: Suggestive contours for conveying shape. *Proc. of SIGGRAPH* (2003), 848–855. 2

[HDS96] HUMPHREY W., DALKE A., SCHULTEN K.: VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics 14* (1996), 33–38. 2

[HPW*12] HUANG J., PEI W., WEN C., CHEN G., CHEN W., BAO H.: Output-coherent image-space lic for surface flow visualization. *IEEE PacificVIS* (2012), 137–144. 2

[Kaj09] KAJALIN V.: Screen-Space Ambient Occlusion. In *ShaderX7 : Advanced Rendering Techniques*. Charles River Media, 2009, pp. 413–424. 2

[KSES12] KRONE M., STONE J. E., ERTL T., SCHULTEN K.: Fast Visualization of Gaussian Density Surfaces for Molecular Dynamics and Particle System Trajectories. In *EuroVis - Short Papers* (2012), vol. 1, pp. 67–71. doi:10.2312/PE/EuroVisShort/EuroVisShort2012/067-071. 1, 2

[LKEP14] LAWONN K., KRONE M., ERTL T., PREIM B.: Line Integral Convolution for Real-Time Illustration of Molecular Surface Shape and Salient Regions. *Computer Graphics Forum 33*, 3 (2014), 181–190. doi:10.1111/cgf.12374. 1, 2

[mev] MeVisLab – medical image processing and visualization. URL: http://www.mevislab.de/. 2

[SOS96] SANNER M. F., OLSON A. J., SPEHNER J.-C.: Reduced Surface: An Efficient Way to Compute Molecular Surfaces. *Biopolymers 38*, 3 (1996), 305–320. 2