Onderzoeksstage - An abstraction space for molecular visualization

Matthew van der Zwan

Abstract—A lot of research revolves around molecules, resulting in very different ways to visualize the corresponding molecular data, focused on showing relevant properties. For proteins, in particular, people tend to use some very different representations.The goal of this research is to provide researchers with a tool in which they can intuitively and interactively explore the structure of a protein, thereby aiding them in their research.To this end, we developed an abstraction space, which allows dedicated control over the structural representation of the protein, the rendering style and the added depth cueing techniques.

Index Terms—Molecular visualization, illustrative visualization, NPR, depth cueing, ambient occlussion, halos, object attenuation

1 Introduction

Visualizing molecular data is important in a vast number of scientific disciplines, like medicine, material sciences and chemistry. People in all of these disciplines have their own favorite way of visualizing their data, even within one discipline a lot of different styles are used. These styles differ in multiple aspects, such as the way the structure of the molecule is represented and the style of the resulting image. The image styles range from photorealistic representations, as often found in scientific journals, to black-and-white illustrations, for instance found in study books. The process of creating these images using already available software consists of guessing a lot of parameters and tweaking them until the result is as intended.

Visualization software is also used to explore the structure of a given molecule. Here, again, different structural representations are used, depending on the (sub-)structure that needs to be identified. For some areas of research, like bio-chemistry, it is helpful to be able to switch between different representations. Besides different structural representations, other techniques are used to emphasize the three-dimensional structure of a molecule. Techniques used to emphasize the depth perception of an object are commonly used in illustrative visualization and are often referred to as depth cueing techniques.

We created a method that allows the interactive selection of the structural representation and image style, both for exploring the overall structure and for creating an image focusing on the region that is most important to the user. This visualization method is based on what we call an abstraction space. Every point in the abstraction space represents a combination of abstractions of different quantities. The goal of this research is to develop an abstraction space in which there exist continuous transitions between every two points in the abstraction space. This way the user can select any level of structural abstraction and illustration style, together with the desired amount of depth cueing, as demonstrated in Fig. 1.

We start with describing work that is related to this research in Section 2. Then, we describe the idea behind our developed technique in Section 3, and elaborate on how this is realized in Section 4, giving the important implementation details in Section 5. After that we give some results in Section 6, which we discuss in Section 7.

2 Related work

A lot of research has been done in the direction of molecular visualization [14]. First, illustrators created hand-drawn illustrations (Fig. 8(a)), while in more recent times, computers are commonly used for creating such images. The images created using a computer can still look like the pen-and-ink renderings as created by illustrators (Fig. 2(a)), but can also aim to allow real-time photorealistic rendering of large molecules (Fig. 2(b)).

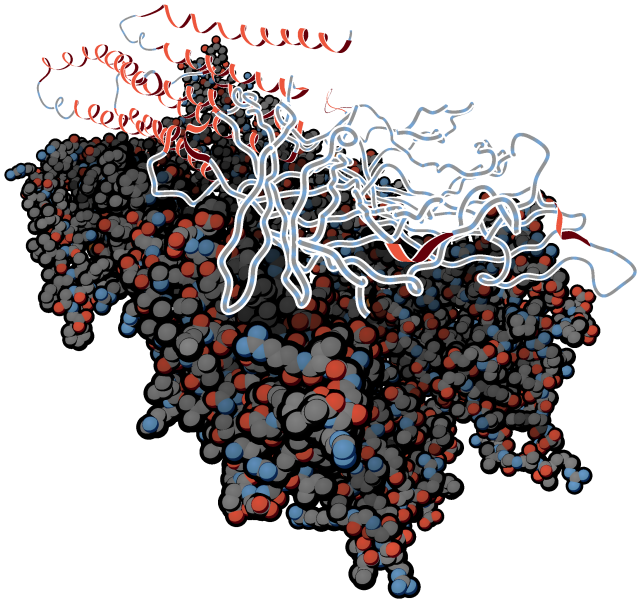


Fig. 1. Pastel colored rendering of a combination of two different structural abstraction styles, with full depth cueing.

The methods mentioned before use techniques from the field of illustrative visualization, either because the aim is to create an illustration [20], or to enhance depth perception in the image [18]. Commonly employed illustrative visualization techniques to enhance depth perception are halos [1, 5, 8], and silhouettes and contour lines [7, 12].

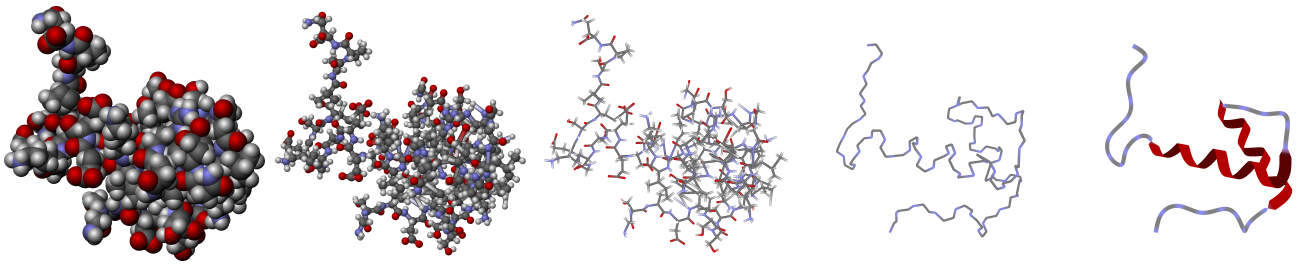
The basis for this method was created during research for the student colloquium [19], but the result of this work had a number of limitations. For example, it only allowed changing the structural representation while changing the visualization style at the same time and had minimal depth-cueing in the form of halos. Also, the transitions were far from smooth, resulting in incoherent animations. And the parser used to load data files worked on only one out of all possible data files, making it very limited in its use.

3 Abstraction Space

The goal of this research is to develop an abstraction space for molecular visualization, which allows users to have dedicated control over the structural depiction and rendering style, allowing the visualization of protein data from the Protein Data Bank [3, 4]. The transitions over each axis should be smooth and completely independent of the other axis. Moreover, we want to provide the user with a simple and intu-

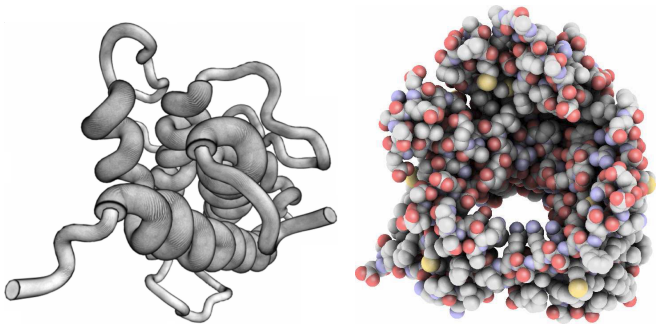
• *The author is a student at the University of Groningen, The Netherlands, E-mail: m.a.t.van.der.zwan@student.rug.nl.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mailto:tvcg@computer.org.



(a) (b) (c) (d) (e)

Fig. 3. Five representations of a small protein with corresponding structural abstraction values. Starting with (a): Space-fill (0.0), then (b): Balls-and-sticks (0.3), followed by (c): Licorice (0.6), (d): Protein backbone (0.75), and (e): Ribbons (1.0).



(a) (b)

Fig. 2. Images created with existing programs/techniques: (a) Pen-and-ink style drawing of the secondary structure, from [20] and (b) Photorealistic rendering using ambient occlusion of the primary structure, from [18].

itive way to add depth cueing, in the form of halos, object attenuation and ambient occlusion.

We start this section by giving a definition of the term abstraction space and describing the properties and requirements that should hold for our abstraction space, in order to create smooth transitions in this space. Then, we describe the type of abstractions we want to employ for our abstraction space for protein visualization.

3.1 Definition

The abstraction space is a space $\mathscr{F}$ of functions. Every function $f(t_s,t_p,t_i) \in \mathscr{F}$ with $t_s,t_p,t_i \in [0,1]$ consists of a function triple $(f_s(t_s),f_p(t_p),f_i(t_i))$, where f_s determines the degree of structural abstraction, f_p the support of spatial perception, and f_i the 'illustrativeness.' Each of these functions f_k has several discrete levels $t_{k,n}$, evenly spaced in $[0,1]$, associated to it that mark known styles. Each f_k needs to specify how to seamlessly transition between $f_k(t_{k,n})$ and $f_k(t_{k,n+1})$. A mapping A assigns to each amino-acid a in the protein a tuple $(t_s(a),t_i(a))$, thus determining its style. The parameter t_p can only be controlled globally.

3.2 Application

For our envisioned application, the abstraction space is formed by three axes:

Structure determines the structural representation
Illustrativeness determine the imaging (and coloring) style
Spatiality determines the applied depth cueing

Therefore, every configuration in the abstraction space is a combination of these three methods of abstraction. The abstraction map we use applies the depth cueing on a global level, while the other two are applied locally, allowing us to set a location specific illustration style and level of structural abstraction, while determining the amount of depth cueing for the entire image.

As primary abstraction, we consider the structural abstraction, that is, the way we represent the protein structure. As representations, we first consider the space-fill (or CPK) model, where we display every atom as a sphere at its van der Waals radius (Fig. 3(a)). Using this representation, we only see the outer surface of the molecule as a union of spheres [6, 9]. The second representation we consider, balls and sticks, shows some more of the internal structure, using smaller spheres and cylinders, which represent the bonds between atoms (Fig. 3(b)) [9]. A better look at the internal structure of the protein is possible by only showing the bonds, again as cylinders, known as the licorice representation (Fig. 3(c)) [9]. The internal structure is even better visible when only viewing the protein backbone [15], where we have left out the less important external structures (Fig. 3(d)). When studying the protein backbone, the individual atoms are of less importance and, therefore, they are abstracted in the last structural representation we use, the ribbon representation [16]. In the ribbon visualization, the atoms that form a helix structure are shown as an abstracted helix, while the other parts of the chain are shown as a ribbon or tube (Fig. 3(e)).

Our secondary abstraction axis concerns the coloring and drawing style of the resulting image. First, we consider photorealistic images, where the atoms are colored according to their atom type and bonds and ribbons are colored according to the colors of the atoms they connect. For the photorealistic imaging style, we further use a 'realistic' lighting model, which does not really exist for structures as small as atoms, but is required to create the illusion of depth in the images. We also envision people using our program to create images to use in books and publications, most of which are printed in black and white [10]. Therefore, we provide the ability to create black and white images, where we use different hatching patterns to distinguish the atom types, but now longer use shading. As an intermediate level of illustrative abstraction, we use pastel colors that are close to the real atom colors, while applying discrete shading.

Our last axis controls the added depth cues, which are added to enhance the perception of depth in the resulting image. To this end, we add halos to the objects, these halos block the objects behind them, emphasizing objects in the foreground [1, 5, 8]. For volumetric objects, like the spheres and cylinders, the structure can be made more apparent by applying ambient occlusion [13, 18]. As a last technique to enhance depth perception, we apply object attenuation, reducing the size of objects which are further away and increasing the size of objects that are close.

4 Realization

Now we have introduced the concept of abstraction space and given a description of the function for each axis, we present the realization of the axes. We explain how to implement the abstraction along a particular axis and how this is coupled with the other axes. All along, we also ensure that we have continuous transitions throughout the entire abstraction space.

4.1 Structural abstraction

For the structural representation, we consider the space-fill representation as the non-abstracted form, that is, we start our structural ab-

straction with this representation. The transition to balls-and-sticks is performed in a straight-forward way, by shrinking the radii of the spheres when increasing structural abstraction, the bonds become visible and we get a balls-and-stick representation of the protein. We can continue shrinking the spheres further, until they have the same radii as the cylinders, resulting in the licorice representation.

The transition from the licorice representation to the protein backbone is less trivial then the previous transitions, because we need a method that allows continuous interpolation between the two representations. To this end, we first introduce a distance measure, which for a given atom defines the distance to the protein backbone. This function is defined as follows:

$$\text{dist}(a) = \begin{cases} 0 & \text{if } a \in \text{backbone} \\ 1 + \min_{b \in \text{conn}(a)}\{dist(b)\} & \text{otherwise} \end{cases}$$

Here, $\text{conn}(a)$ are all the atoms which are connected to atom a. This function correctly assigns the largest distances to the atoms furthest away from the protein backbone and decreases when we pick atoms closer to the backbone. However, since the transition starts from licorice visualization, we are no longer concerned with displaying atoms. Therefore, we use this definition of backbone-distance for atoms to decide which bonds we should render. For this purpose, consider a bond connecting two atoms a_0 and a_1, such that $\text{dist}(a_0) < \text{dist}(a_1)$, and a non-integer threshold T, the latter based on the required level of structural abstraction. When both atoms are further away from the backbone then our threshold value (i. e. $\text{dist}(a_0) > T$, we do not render the connecting bond. Likewise, if both atoms are closer to the backbone then the threshold value (i. e. $\text{dist}(a_1) < T$), we render the entire bond. In the last case, where the threshold value is between the distance values of the atoms (i. e. $\text{dist}(a_0) < T < \text{dist}(a_1)$), we only render that part of the bond where the distance is smaller than the threshold value.

In order to move from the backbone representation (where we still have the linear structures of the bonds) to the ribbon representation (which has a smooth, non-linear character) we interpolate a smooth function through the atoms which are part of the backbone. During the transition from backbone to ribbon representation, we apply a linear interpolation between the linear backbone and the smooth function. At the same time, we increase the width of the lines, while changing the orientation for those parts of the ribbons that make up helices. This is because the lines in ribbon representation are generally orientated such that they are facing the viewer, whereas they are oriented as if they are wrapped around an internal cylinder for the helices. For our goal of continuous transitions we cannot just change the orientation, so we apply another interpolation for this transition in orientation.

4.2 Illustrativeness abstraction

As with the structural abstraction, we start by defining the starting point of our illustrativeness abstraction, which is the photorealistic rendering style (Fig. 4(a)). In this style, we color atoms, bonds, and ribbons according to the atom type, as explained before. For the ribbons, we make an exception for the helices, which we give a different color to make them easier to identify. Furthermore, we apply a continuous darkening towards the edges of atoms and bonds to emphasize their volumetric appearance. For the helices, we darken the inward facing parts of the helix to make them more distinguishable from the outward facing parts, again giving better spatial perception then using flat colored regions.

The second rendering style on our illustrativeness axis is the pastel colored rendering (Fig. 4(b)), where we use two pastel colors, one light and one dark color, and a simplified shading model. This shading model is based on the shading model used for the photorealistic rendering, but instead of a continuously darkening shadow, we introduce a threshold. When the illumination is below the threshold, we use the dark pastel color, otherwise we use the light color. For the helices, we explicitly use the dark pastel color for the inward facing parts and the light pastel colors for the outward facing parts, similar to the photorealistic rendering.

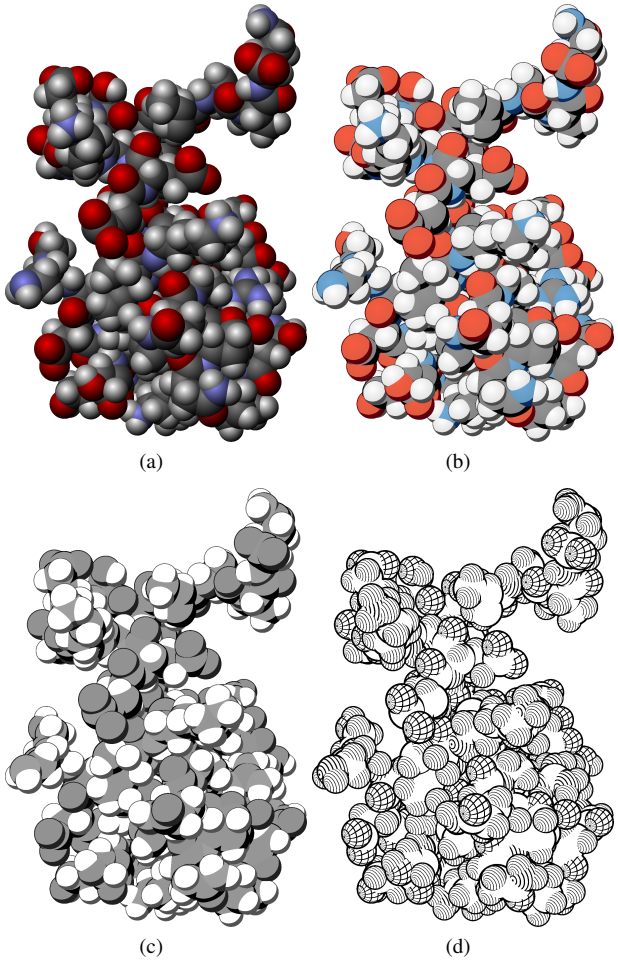


Fig. 4. Several stages along the illustrativeness axis for a space-fill rendering. Starting with (a): Photorealistic, then (b): Pastel colors, (c): Greyscale, and (d): Black and white hatched illustrative style.

At the end of our illustrativeness abstraction axis, we have the black-and-white illustrations, therefore we want a transition from the pastel colors to the hatched black-and-white images. A smooth transition is ensured by adding an extra illustrative style in between, using grey tones based on the pastel colors (Fig. 4(c)). The greyscale values are created by desaturating the pastel colors and therefore the resulting image has two tones and discrete shading. Now we have a greyscale image, the transition to a black-and-white image, where we use hatching patterns to indicate the type of atoms or orientation on the helices (Fig. 4(d)), looks much smoother. Now we have defined how we create all illustration styles, we only need to specify how to realize smooth transitions. These transitions are created by linearly interpolating the colors between two consecutive illustration styles.

4.3 Spatiality enhancement

In contrast with the previous two axis, the spatiality enhancement axis is not an interpolation between different structural representations or rendering styles, but is used to enable depth cueing techniques. This leads to a different problem for the realization of this axis, because the sequence in which to apply these techniques is not as apparent as the sequence of abstractions for the previous two axes. The sequence in which the depth techniques are applied when moving along the spatiality axis, is chosen based on the properties of the individual depth cueing techniques. As an example, ambient occlusion has been proven as an effective way to enhance depth perception for space-fill and balls-and-sticks models. For the sparse structures of the ribbon view, on the other hand, light will be able to reach almost all places

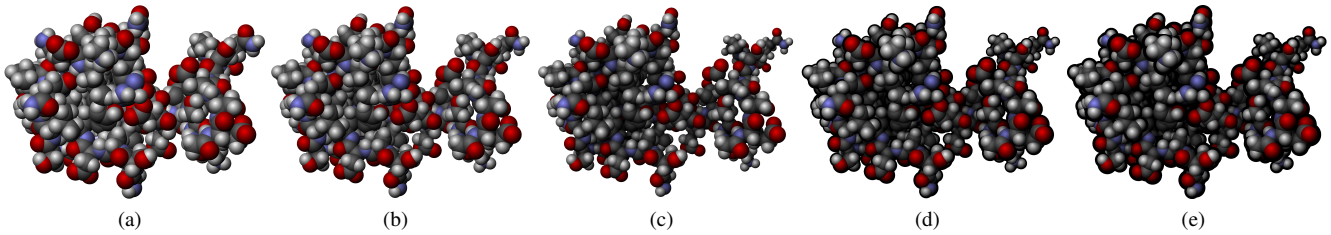


Fig. 5. Several stages along the spatiality axis for a space-fill represented molecule. Starting with (a): No depth cueing, then (b), (c): Some, resp. full added ambient occlusion and object attenuation, and (d), (e): Some, resp. full added halos.

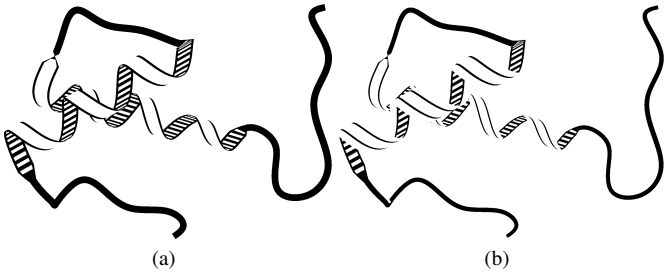


Fig. 6. A black-and-white rendering of a ribbon represented molecule. (a): No depth cueing, (b) Full object attenuation and halos.

and ambient occlusion will not enhance depth perception. The other techniques (object attenuation and halos) can both be applied for all levels of structural abstraction, but because halos around objects block objects behind them, we lose some of the effects of the other techniques. Therefore, we chose to apply halos as last depth cueing technique and apply ambient occlusion and object attenuation at the same time, to avoid regions on our spatiality enhancement axis where nothing happens for certain levels of structural abstraction.

Now we have determined the sequence in which to apply the depth cueing techniques, we can show the results on the space-fill representation of our demonstration molecule (Fig. 5(a)). First, we add ambient occlusion and apply object attenuation (Fig. 5(b) and 5(c)). After that, we apply halos (Fig. 5(d) and 5(e)), where we see that objects further to the back are blocked by the halos. We also applied the depth cueing techniques for a ribbon view of the same molecule in black-and-white illustrativeness abstraction to demonstrate the benefit of the added depth cues (Fig. 6).

5 Implementation

Most of the methods that together form our abstraction space can be implemented in a straight-forward way, but for some parts we have to take more care. Because we want the program to be interactive, we also need to take the performance of the implementation into account, which we do by implementing the techniques using shaders an vertex buffers. Furthermore, we use impostors to render the sphere and cylinders in, which we describe in Section 5.1. After that, we conclude this section by describing an efficient way to store the colors for all atom types in Section 5.2.

5.1 Impostors

When rendering a molecule in balls-and-sticks representation, we depict the atoms as spheres and the bonds as cylinders. Because all rendering primitives are based on triangles, we need a lot of these triangles to approach the smooth surface of a sphere or cylinder. The amount of triangles that are rendered each frame is a big factor in the performance of a program. Furthermore, we have to make the transition from the cylindrical bonds in the molecule skeleton, which have volume, to the flat lines and ribbons used in the ribbon visualization. For these reasons, we use the sphere and cylinder impostors described by Tarini et al. [18] and Bajaj et al. [2] to render the atoms and bonds.

The impostors we use are planes which are oriented towards the user, on which we draw the shape of the object it should represent. To make sure that it behaves as the real object does, we apply a depth offset according to the shape of the real object [17]. This way, we need only two triangles to represent the smooth shapes of the spheres and cylinders, while the transformation to the flat lines and ribbons is much easier then with the real objects. To transform from a cylinder impostor to a line, we have to change what is drawn on the plane and undo the depth offset.

Besides describing the impostors, Tarini et al. also describe how to calculate ambient occlusion using these impostors, we follow their approach when doing so. To calculate the amount of light reaching an atom, we render the scene from all the light sources and determine which atoms can be seen. If an atom can be seen from the light source, we increase the amount of light for that atom, which is stored in a texture. After going through all light sources, this texture holds the amount of light reaching all atoms and can be used to give them the correct lightness (or darkness) during rendering. Because the molecule and position of the light do not change, we calculate this texture during a pre-processing stage.

5.2 Colormap

As explained in Section 4.2, we use three different colors per atom type, one photorealistic color and two pastel colors. Since each color is represented by three floating point numbers (floats), we need to store a total of nine floats for every atom we display, for color storage alone. For bonds we need to store even more information, because a bond connects two atoms. This results in twice the amount of color information required for atoms, leading to a total of 18 floats to store the color information of a bond. If we can reduce the amount of information that has to be transferred each frame, we might be able to improve performance.

However, performance is not our only concern when it comes to the colors of the atoms. We also want to allow the users to choose which colors they want to use for the different types of atoms. Therefore, we introduce a colormap, which stores the colors used for all atom types. Since this colormap only has to change when we want to use other colors, we store it in a texture, while we use a texture coordinate to indicate the atom type. Now we only have to store this texture coordinate for every atom in the molecule. This leads to a reduction of 8 floats which we have to store per atom and, consequently, a reduction of 16 floats for the bonds, where we now only have to store two texture coordinates. Unfortunately, this reduction does not lead to a performance gain, allthough there is no performance loss either, but we do provide the user with a method to change the color scheme. However, the reduction in memory usage does allow for the storage in video memory and therefore, the rendering of larger proteins.

6 Results

Now we have discussed the principles, realization and implementation details of the abstraction space for protein visualization, we present some results in this section. We first present some graphical results and conclude with performance measures.

All images throughout this paper are created using our reference implementation of the described techniques. Different forms of structural

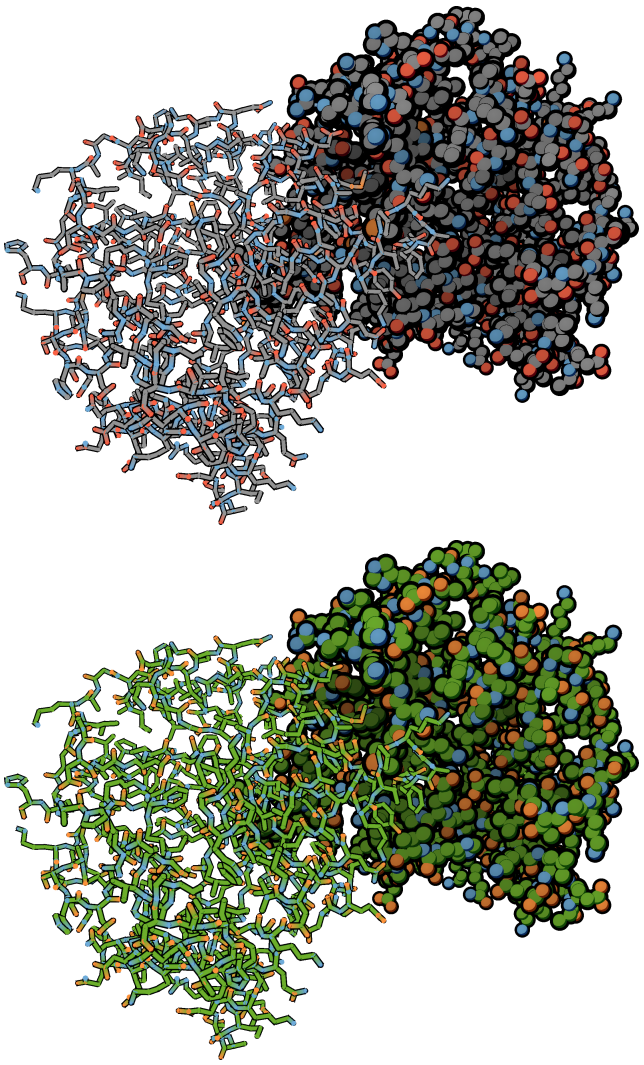


Fig. 7. The use of two different colormaps to represent the same molecule using the same levels of structural abstraction and seen from the same point of view. The top image uses the default colormap, which uses conventional colors, while the bottom image uses an alternative colormap.

abstraction and combinations thereof are found in Fig. 1 and Fig. 3, whereas Fig. 4 shows the different levels of illustrativeness, with Fig. 6 showing an illustrated ribbon representation. The effect of the depth cueing techniques is shown in Fig. 5 and Fig. 6.

The result of using different colormaps is shown in Fig. 7, where we again see a combination of two different levels of structural abstraction. Using the alternative colormap seems to enhance the effect of the dark halos, because of the bigger contrast compared to the original colormap. This might also be useful to use on a more local scale, at least for ribbons, where we might want to emphasize one particular strand in a big protein, which we could do by changing the colormap for that strand to make it 'pop out' more.

Besides creating photorealistic images, we also aim at creating black-and-white images to be used in books or articles. Fig. 8 compares two sets of images, images created by a professional illustrator and images created using our technique. Allthough there is still room for improvement, the images we created look good enough to be used in a publication. The addition of directional cues for the ribbons, as the illlustrations have, might add to the overal quality of the images.

We measured the performance of our technique by running a reference implementation and determining the frame rates at both ends of the structural abstraction axis, while changing the other abstraction settings. We ran this test on an Intel Core 2 Quad running at 2.4 GHz, equipped with 3 GB of memory and a Geforce 8800GTS with 512 MB of memory, running 64-bit Microsoft Windows 7 with the most recent drivers. The results of this test can be found in Table 1, where we see that we retain interactive frame rates even we have about 60,000 atoms in a protein. For 12,000 atoms and less we even get real-time frame rates. We also see that the frame rates are higher when using full structural abstraction, ranging between a factor 3 to more then a factor 4.

Fig.	atoms	$bonds_p$	$bonds_s$	FPS_0	FPS_1
2-4	796	152	647	279	1228
5	7343	2794	4668	130	490
1	12605	4645	8300	83	344
1AON	58674	24024	35063	22	79

Table 1. Performance measures taken for the molecules also used in the indicated figures. Here atoms is the number of atoms, $bonds_p$ and $bonds_s$ the number of bonds inside resp. outside the molecule skeleton. The number of frames are measured for no (FPS_0) and full structural abstraction (FPS_1).

7 Discussion and Future Work

The goal of this research was the development of a continuous abstraction space for protein visualization. We succeeded in doing this, allowing users to interactively determine the settings they want to use for their rendering on a per axis fashion. This allows abstraction over the structural axis, without drastical changes in the rendering style, as is common in existing tools [20]. The gradual addition of the depth cueing techniques allows for selecting the right amount of depth cueing for the current molecule without presenting the user with a dozen parameters. Overall, we achieved the goals we set for ourselves and have managed to create a program which allows dedicated control over the abstraction space for molecular visualization.

The techniques developed during this research can also applied for molecules that are no proteins, allthough, given the current realization of the structural abstraction axis, the structural abstraction then stops with licorice. As the current implementation is build on the assumption that we deal with protein data, it will have to be adapted to deal with non-protein data. The loading of proteins has been greatly improved over the previous version, since it now allows a wide variety of proteins to be loaded, but it still needs improvement to be able to load proteins that are large than the ones used in this paper.

During a demo session with some people from the chemistry department, they told us they see real potential for this technique, although they would like to see some adjustments. For one, the proteins they work with are a lot bigger then the ones we use, so we will need to work on boosting the performance to be able to render more complex structures in real time. The results of there simulation are not stationary single proteins, but multiple proteins that interact with each other over the simulated timespan. We, therefore, have to adopt our technique to be able to cope with this time-axis, perhaps expanding our abstraction space to 4 dimensions, the fourth dimension being how we abstract the timeline.

Representing large proteins is already a challenge by itself, even without the time-dependence, as we saw in our performance measure. The large proteins used by chemists the days, can probably not be visualized at interactive frame rates using our method. We already saw that higher structural abstraction of the molecules leads to an increase in frame rate, therefore it might be worthwhile to investigate representations of coarse grained molecules, like the MARTINI model. When studying these large datasets, the solvents which are present in the dataset are of interest as well. Providing a way to effectively show these solvents and their interaction with the protein(s) is of importance to the field of molecular dynamics too.

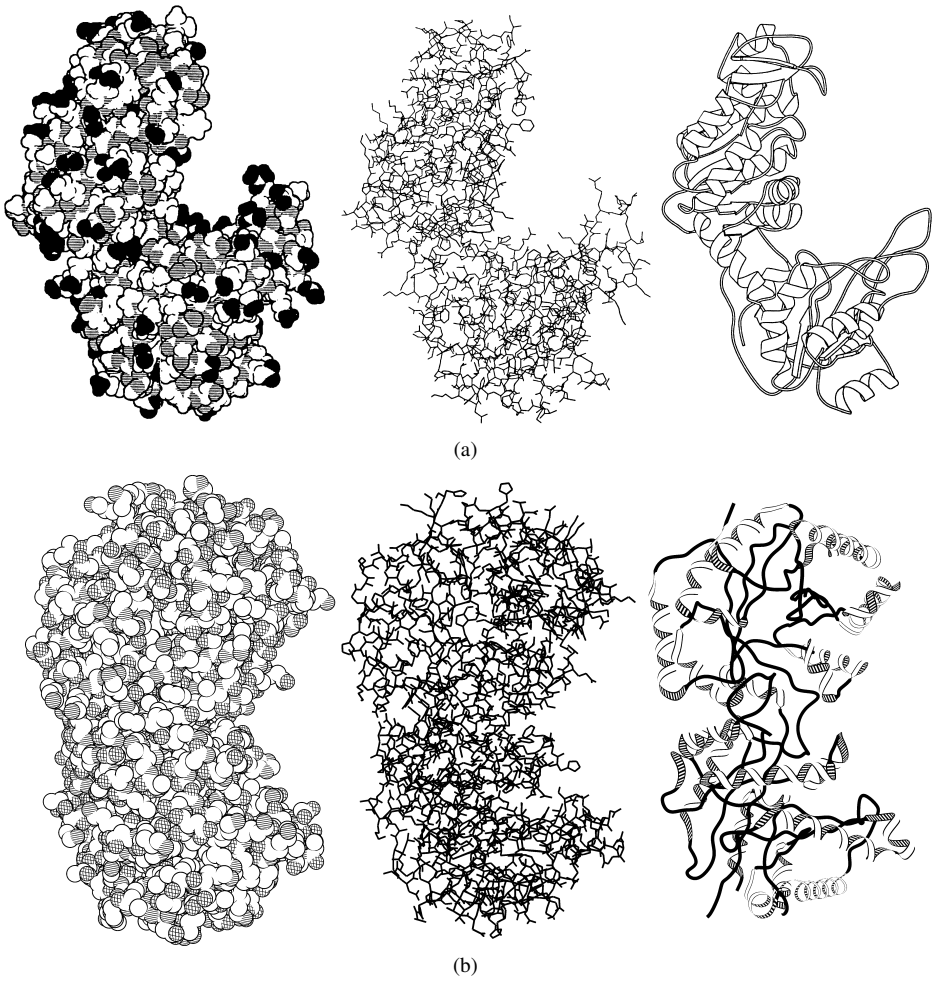


Fig. 8. Comparison of images created by a professional illustrator (top row, from [11]) and images created using our program (bottom row). From left two right the molecule is depicted using the following representations: space-fill, licorice and ribbon.

REFERENCES

[1] A. Appel, F. J. Rohlf, and A. J. Stein. The haloed line effect for hidden line elimination. In *SIGGRAPH '79: Proceedings of the 6th annual conference on Computer graphics and interactive techniques*, pages 151–157, New York, NY, USA, 1979. ACM.

[2] C. Bajaj and P. Djeu. Texmol: Interactive visual exploration of large flexible multi-component molecular complexes. In *In VIS 04: Proceedings of the conference on Visualization 04*, pages 243–250. IEEE Computer Society Press, 2004.

[3] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, Jan. 2000.

[4] H. M. Berman, J. Westbrook, Z. Feng, T. N. Gilliland, G. Bhat, H. Weissig, I. N. Sindyalov, and P. E. Bourne. The Protein Data Bank. *Acta Crystallographica Section D*, 58(6 Part 1):899–907, Jun 2002.

[5] S. Bruckner and E. Gröller. Enhancing depth-perception with flexible volumetric halos. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1344–1351, 2007.

[6] R. B. Corey and L. Pauling. Molecular models of amino acids, peptides, and proteins. *Review of Scientific Instruments*, 24(8):621 –627, Aug. 1953.

[7] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *ACM Trans. Graph.*, 22(3):848–855, 2003.

[8] M. H. Everts, H. Bekker, J. B. Roerdink, and T. Isenberg. Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1299–1306, 2009.

[9] D. S. Goodsell. Visual Methods from Atoms to Cells. *Structure*, 13(3):347–354, Mar. 2005.

[10] D. S. Goodsell and A. J. Olson. Molecular Illustration in Black and White. *Journal of Molecular Graphics*, 10(4):235–240, Dec. 1992.

[11] E. R. S. Hodges, editor. *The Guild handbook of scientific illustration*. John Wiley & Sons, 2003.

[12] T. Isenberg, B. Freudenberg, N. Halper, S. Schlechtweg, and T. Strothotte. A developer's guide to silhouette algorithms for polygonal models. *IEEE Comput. Graph. Appl.*, 23(4):28–37, 2003.

[13] J. T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, 1986.

[14] D. R. Lipsa, R. S. Laramee, S. J. Cox, J. C. Roberts, and R. Walker. Visualization for the Physical Sciences. In *Eurographics 2011 State of the Art Reports*, number Aire-la-Ville, Switzerland. Eurographics Association, 2011. To appear.

[15] K. Namba, D. L. D. Caspar, and G. Stubbs. Enhancement and Simplification of Macromolecular Images. *Biophysical Journal*, 53(4):469–475, Apr. 1988.

[16] J. S. Richardson. Schematic Drawings of Protein Structures. In *Diffraction Methods for Biological Macromolecules Part B*, volume 115 of *Methods in Enzymology*, pages 359–380. Academic Press, 1985.

[17] G. Schaufler. Nailboards: A rendering primitive for image caching in dynamic scenes. In *Proceedings of the Eurographics Workshop on Rendering Techniques '97*, pages 151–162, London, UK, 1997. Springer-Verlag.

[18] M. Tarini, P. Cignoni, and C. Montani. Ambient occlusion and edge cueing for enhancing real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1237–1244, 2006.

[19] M. van der Zwan and W. Lueks. Depth Cueing and Haloing for Molecular Visualization. In *SC@RUG 2010 proceedings*, pages 17–22, 2010.

[20] J. Weber. ProteinShader: illustrative rendering of macromolecules. *BMC Structural Biology*, 9(1):19, 2009.