CarboHydra: Rendering Carbohydrate Cartoons

Ian Eborn
ieborn@cs.uct.ac.za

Anton Burger
aburger@cs.uct.ac.za

Michelle Kuttel
mkuttel@cs.uct.ac.za

James Gain
jgain@cs.uct.ac.za

Technical Report CS04-18-00
Department of Computer Science
University of Cape Town

ABSTRACT

Molecular visualisation algorithms present simplified views of molecules, which allow chemists to gain insight into their structure and function. Many such algorithms are currently in use, including the ribbon diagrams by Richardson [12], which were developed specifically for viewing proteins. None are customised to the visualisation of carbohydrates – in particular, the rings of atoms formed within them. We present two novel algorithms (PaperChain and Twister) for this purpose. Both algorithms identify these ring structures, and attempt to convey a variety of their properties. We have implemented a molecular viewer which displays carbohydrates using the two novel algorithms, as well as the conventional ball-and-stick view. The system permitted comparative tests of the three algorithms by research chemists, as well as evaluation of the effectiveness of a number of rendering options.

1. INTRODUCTION

The field of molecular visualisation is concerned with presenting abstractions of molecules which convey selected aspects of their structure. Such information can be interpreted by chemists to provide information about their chemical properties and function. Despite the many molecular visualisation algorithms in use, none cater specifically to the visualisation of carbohydrates.

We present two novel algorithms to address this shortcoming. The algorithms are intended to convey information about the layout, orientation and shape of the ring substructures found within carbohydrates. The PaperChain algorithm focuses on depicting the conformation of individual rings, while the Twister algorithm attempts to depict the differences in orientation between connected rings. The algorithms have been implemented in a customised molecular viewer, which uses OpenGL to display molecules using the novel algorithms as well as the more traditional ball-and-stick algorithm. The system includes a number of rendering options, such as texture mapping, a custom lighting model and silhouette edges. We evaluate the effectiveness of the novel algorithms, with and without these features.

2. BACKGROUND

The prior knowledge required for the design and implementation of the system can be divided into three areas: visualisation, the structure of carbohydrates and rendering features.

2.1 Visualisation

2.1.1 Visualisation principles

Visualisation of data is achieved by representing it as a collection of *marks* [13]. A mark is a graphical primitive such as a point, line or area. The information to be conveyed is encoded in various properties of marks, which can be divided into three categories. Positional properties include position, size and orientation. Temporal properties affect the changes of other properties over time. Retinal properties include all other properties detectible by the human visual system, and include colour, texture and shading. This rigorous classification of the elements of a visualisation algorithm allows the application of rules to increase an algorithm's effectiveness at conveying the intended information. These rules are typically formulated as a result of tests with users, and take the form of statements such as "Transparency allows areas of high interest to be spotted quickly by looking through areas of low interest" [13].

2.1.2 Molecular visualisation

Well-known molecular visualisation algorithms include the ball-and-stick and CPK models. Ball-and-stick simply models each atom as a small sphere, with the bonds between atoms represented as "sticks", typically thin cylinders. CPK models each atom as a sphere, whose radius is the atom's van der Waals radius. This can be useful in predicting the nature of interactions between molecules.

Both the CPK and the ball-and-stick models have the problem that the structure of the molecule is not readily appar-

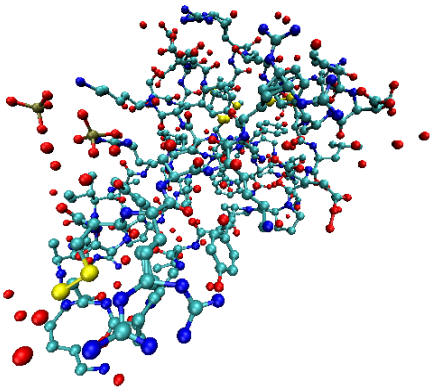

Figure 1: An illustration of the ball-and-stick model, clearly showing the difficulty in discerning detail when viewing large molecules.

ent; explicit representation of all atoms results in an image that is cluttered and difficult to interpret, especially for large molecules (see Figure 1, generated in the popular molecular dynamics package, VMD [9]). The hand-drawn ribbon diagrams invented by Richardson [12], later extended to make use of computer graphics by Carson and Bugg [6, 5], alleviate this problem by producing more abstract views of molecules. Only the backbone of the molecule is represented, allowing important structural features to be clearly seen.

2.2 Carbohydrate structure

A significant feature of carbohydrate polymers are the connected rings of atoms from which they are formed. The rings are not planar, instead assuming various non-planar conformations dependent on their energy state, e.g. "chair" or "boat" conformations. The carbon atoms of the rings are numbered sequentially according to a convention maintained by the International Union of Pure and Applied Chemistry [10]. This numbering allows the definition of a "top" and "bottom" side of each ring, resulting in a consistent way of defining a ring's orientation.

2.3 Rendering features

2.3.1 Technical lighting model

The objective of molecular visualisation is not realistic representation, but rather the indication of the conformation of the molecule. This means that it is more important for a visualisation algorithm to indicate all of the features of the visualised molecule clearly than to have a realistic lighting model. In fact, a realistic lighting model could be detrimental to an algorithm, as shadows can hide important details.

The technical lighting model proposed by Gooch, *et al.* attempts to preserve the geometric detail of objects in shadow [8]. Where traditional lighting shades using a luminance scale, using light and dark shades to suggest illumination, their technique partially replaces the luminance scale with a tonal scale, using cool colours for dark areas, and warm colours for lighter ones. By avoiding the use of extremely dark or light colours, parts of objects that would otherwise be obscured by shadows or highlights remain visible. This also leaves black free for use in rendering silhouettes.

2.3.2 Silhouette rendering

Research in cognitive psychology suggests that the presence of an object's silhouette in an image is important in determining the shape of the object [3]. Previous work in rendering silhouettes includes Buchanan and Sousa's edge buffer [4], in which a data structure stores a collection of bits for each edge in a polygon mesh. As faces are rendered, different bits for each of their edges are toggled between 0 and 1 depending on the orientation of the face relative to the viewer. The end result is a classification of each edge, allowing a decision to be made on whether to render it or not.

Raskar and Cohen propose an image-space approach to rendering silhouettes [11]. While less flexible than the edge buffer (only silhouette edges are rendered), the technique does not rely on manipulation of potentially large data structures. Using the fact that silhouette edges are defined as the intersection of front and back faces, their solution overwrites the front faces of objects with the back faces, but only at pixels whose depth values for the two types of polygon are the same. The back faces are rendered in the desired silhouette colour.

3. APPROACH

3.1 Molecular files

The first stage in developing the system was the loading and interpretation of molecular definition files, to produce an in-memory graph structure representing a molecule. The Protein Data Bank file format [2] was chosen, as it is widely used and well documented. The minimum required input is a list of atoms with their atomic symbols and rectangular coordinates in Angstroms (metres^{-10}). If atomic connectivity information is present, it is used to construct bonds between nodes (atoms) in the graph. The system infers missing connectivity information by finding pairs of atoms whose distance from one another is less than the sum of their van der Waals radii. Each atom is bonded to the closest neighbour satisfying this requirement, until it has the correct number of bonds. Comparison of the resulting structure with the same molecule viewed in VMD suggested that this technique was an acceptable compromise (see Figure 2 for an example). The completed graph structure was passed to the visualisation modules for processing.

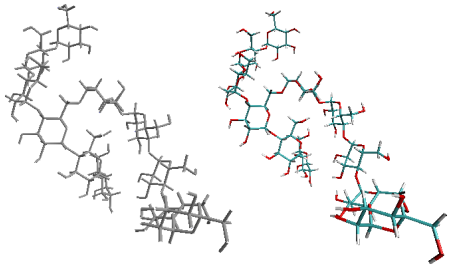

Figure 2: A carbohydrate whose file contained no connectivity information, as loaded by CarboHydra (left) and VMD (right).

3.2 The PaperChain algorithm

The PaperChain algorithm has been designed to show the conformation of each ring in a carbohydrate. It accomplishes this by constructing a non-planar polygonal approximation to the ring, whose vertices are located at the positions of the atoms in the ring. It depicts the linkages between rings by thin cylinders, placing small pipes at the endpoints of each as approximate normal vectors of the rings it joins.

The algorithm starts by finding cycles in the molecule graph, interpreting them as rings. The search algorithm starts at an arbitrary atom, and recursively explores each of its bonds in a depth-first fashion, maintaining a list of visited atoms and bonds. When an atom is visited for the second time, the composition of the cycle is recorded as a ring.

The linkages between rings are also identified using a depth-first approach. A search is started at each atom on each ring, and terminates when it finds an atom on any other ring. The search is not permitted to explore different atoms in the originating ring, simplifying the identification of the endpoints of the linkage.

The geometry for a ring is constructed by calculating its centroid, along with an approximate normal vector using Newell's method [14]. The centroid is offset in both directions along this normal, and the offset points are joined to points at the ring atoms' locations to form triangular faces on both the top and bottom sides of the ring. The resulting closed object preserves the original locations of the atoms in the ring. The oxygen atom in the ring is depicted as a small sphere, to assist in identifying the ring orientation.

The linkages between rings are depicted as thin cylinders. The cross product of an arbitrary vector with the vector between the endpoints yields a third vector, orthogonal to both. Care is taken to ensure that the initial two vectors are not parallel. Using the coordinate frame defined in this way, a circular profile is placed at opposite ends of the vector joining the rings, and orthogonal to it. Corresponding sample points on the profiles are connected appropriately to form triangular faces. Similar, shorter cylinders are generated at the linkage endpoints. These point in the direction of the approximate ring normals, and depict the difference in orientation between the rings at each end of the linkage. The geometry is completed by assigning colours, normal vectors and texture coordinates to each vertex. An example of the PaperChain algorithm, including the use of the technical lighting model and the "silhouette-only" rendering option, is given in Figure 3.

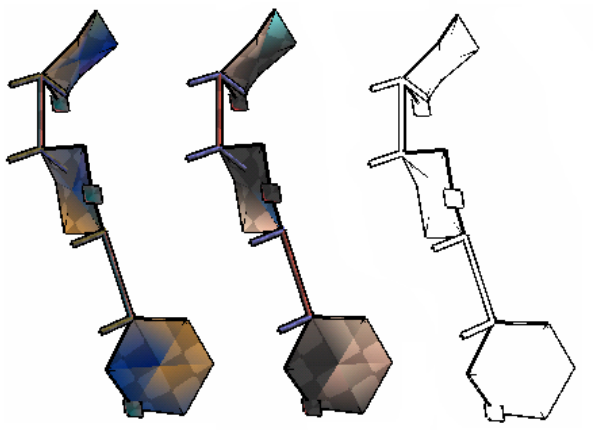

Figure 3: The PaperChain algorithm. The left image uses the technical lighting model. The middle image uses Phong lighting. The right image illustrates the "silhouette-only" rendering option.

3.3 The Twister algorithm

The Twister algorithm has been designed to highlight the relative orientation of rings. It does so by depicting each ring as a thin cylinder. The top and bottom circular faces of the cylinders are assigned two different colours, to allow them to be distinguished. The linkages between rings are represented as ribbons with rectangular cross sections. Their top and bottom sides are coloured in the same way as the cylinders, and they twist through space to align each endpoint with its respective cylinder.

The algorithm limits itself to searching for rings which contain an oxygen atom. It traverses the list of atoms in the molecule, stopping at each oxygen found. If the atom is bonded to two carbon atoms, it is potentially a member of a ring, and a breadth-first search is initiated at one of the carbons, with the other as its destination. Breadth-first search was chosen for its guarantee to find the shortest cycle first. The limited number of branches at each node, and the existence of a maximum practical ring size of 20 atoms both serve to reduce the normally high space requirement for this search type. If the search is successful, a cycle has been identified, and the composition of the ring is recorded. The list of ring atoms is reversed, if necessary, to make traversal of each ring consistent with the IUPAC definition of ring orientation.

Once all of the rings have been found, the algorithm searches for linkages between rings. A limited breadth-first search is started at each atom on each ring, stopping once it finds an atom from another ring. The search is not permitted to explore different atoms in the originating ring, simplifying the identification of the endpoints of the linkage and reducing the number of nodes explored.

Construction of the cylinder for a ring begins in the same way as for the PaperChain algorithm, with the calculation of the centroid, as well as an approximate normal vector using Newell's method. These form the basis for a coordinate frame, in which a circular profile is sampled and translated in both directions along the normal to produce the circular faces of the cylinder, which are divided into triangles for rendering. The cylinder is completed by constructing the side faces and assigning colours, normal vectors and texture coordinates.

Each ribbon is constructed by calculating a cubic Hermite spline between its endpoints. The spline is sampled at uniform intervals, with a coordinate frame calculated at each sample point. The initial frame is defined by the starting point, spline tangent and the normal vector to the starting disk. Each successive frame is calculated using the rotation-

minimising approach detailed by Bloomenthal [7], in which the previous frame derives the new one by being rotated about an axis defined by the cross product of their respective tangent vectors. This approach was chosen to avoid the problems with the Frenet frame, which ceases to exist at points of zero curvature. Each frame is rotated about its tangent by an amount proportional to its distance along the spline, in order to align the final frame with the terminal disk. Triangular faces are constructed between consecutive frames, and assigned colours, normal vectors and texture coordinates. The output of the visualisation modules is sent on to the GUI for rendering. User-specifiable parameters allowed a "pure ribbon" form of the algorithm, in which cylinders were not generated, and separate ribbons met at the ring centroids. Figure 4 provides some examples of the Twister algorithm, using the technical lighting model and "silhouette-only" mode (described in the next section).

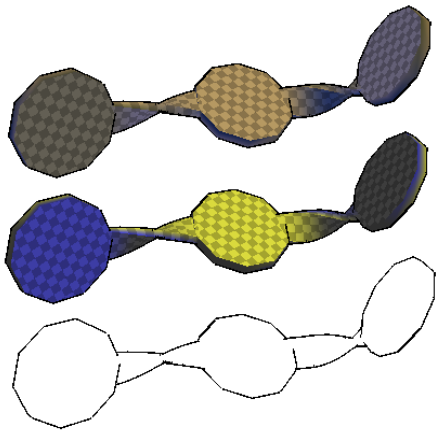

Figure 4: The Twister algorithm. The top image uses the technical lighting model. The middle image uses Phong lighting. The bottom image illustrates the "silhouette-only" rendering option.

3.4 Rendering

The GUI was implemented in the portable, free FOX GUI toolkit [15]. A multiple document interface (MDI) format was implemented, to allow visual comparisons between the output of different algorithms by loading the same molecule multiple times. The interface for manipulation of molecules follows the virtual trackball paradigm [1], with rotation, scaling and translation all accomplished using simple mouse movements.

The interface allows control over the use of texture mapping, technical lighting, silhouettes and the choice of visualisation algorithm. In addition, a translucent ball-and-stick model can be toggled on and off when using one of the two novel algorithms. A "silhouette-only" mode was also implemented, in which the back faces of objects were rendered in thick black lines, with front faces rendered over them in the background colour of the scene. All of this functionality facilitated testing of the algorithms and rendering options.

The technical lighting model was implemented using the approximation suggested by Gooch, *et al.* [8]. The approximation uses two standard OpenGL lights, one each for the warm and cool colours, and allows the technique to be implemented without the need for programmable graphics hardware.

The silhouette edge rendering was accomplished by rendering the back faces of polygons as thick black wireframes, and rendering the front faces over them. The silhouettes are visible beyond the edges of the front faces. This technique deviates from the one presented by Raskar and Cohen, in which the thickness of the silhouettes is achieved by altering the depth values of polygons in the z-buffer, either "pulling" back faces closer or "pushing" front faces further from the viewer. This functionality is achievable in OpenGL using polygon offset. Having been initially implemented, this approach was discarded when it was discovered that no single set of parameters for polygon offset yielded acceptable results across all three visualisation algorithms, producing either near-invisible silhouettes or z-fighting artefacts in thin geometry.

4. RESULTS

The system was tested with research chemists according to the "expert evaluation" paradigm, in which users conversant with the problem domain – in this case, molecular visualisation – are asked to evaluate and provide feedback on a system. Four subjects were tested, and while this is not a large enough group to establish statistical significance, this was deemed relatively unimportant, as the objective was a qualitative estimate of whether further work on the novel algorithms is justified.

Subjects were given a demonstration of the system, and then encouraged to interact with it, filling in a questionnaire as they did so. The questionnaire asked them to rate the visualisation algorithms and the rendering options for each of a number of sample molecules. Subsequent questions asked which algorithm they preferred for viewing large and small molecules, respectively, as well as which they would prefer for producing images for publication.

Subjects' responses were positive; the most significant result was that both novel algorithms were preferred to the ball-and-stick model.

- In general, subjects thought that PaperChain was better at depicting small molecules, or zoomed-in parts of larger ones, as it preserved the conformation of the rings.
- Subjects said that Twister was better at depicting large molecules, as it clearly showed the difference in orientation between connected rings.
- A subject expanded on the preceding two comments by saying that PaperChain depicted secondary structure well, while Twister conveyed tertiary structure. Secondary structure refers to the significant groups of atoms that form within a molecule, while tertiary structure refers to the layout of these groups with respect to one another.
- None of the subjects liked the technical lighting model. Its tendency to distort the colours of objects rendered

the coding of different parts of objects by colour ineffective.

- Most subjects thought the silhouettes were effective at making images clearer. The "silhouette-only" rendering option was deemed useful for producing images for publication, as it looked similar to many sketch-type drawings of molecules used in books.
- Most subjects thought the texture mapping was useful, with one subject saying that it was essential to properly visualise the twisting of the ribbons in Twister.
- Most subjects thought that the ball-and-stick overlay was a useful tool for providing context when viewing a molecule using one of the novel algorithms.
- Several subjects indicated that the Twister algorithm provided insufficient information about the rings – specifically, that it was impossible to identify the bond points of the linkages between rings.
- One subject thought that rendering silhouette edges alone in Twister was insufficient to depict the twist in the ribbons, suggesting that some internal edges also be rendered.
- Several subjects suggested that the pipes indicating normal vectors in the PaperChain algorithm be represented differently, as they looked too similar to atomic bonds.

5. CONCLUSIONS

User testing has established that the PaperChain and Twister algorithms have merit, and that their continued development is worth an investment in time. The manner in which the PaperChain algorithm emphasises individual rings and illustrates their conformation makes it useful as a tool for visualising secondary structure of carbohydrates. The Twister algorithm's ability to clearly depict changes in orientation between connected rings has been deemed useful as a tool for depicting tertiary structure.

Test subjects' comments on the usefulness of seeing the silhouette edges of objects further validate previous research on the subject. The technical lighting model unfortunately did not live up to expectations, but might still find application in situations in which the use of distinguishable colours is not as important.

6. FUTURE WORK

The lack of context in the Twister algorithm might be addressed by colouring different regions of the cylinders according to the atoms which constitute the underlying rings. This is similar to the way in which many implementations of the ribbon algorithm allow pieces of the ribbon to be coloured by the particular amino acids at each point on a protein backbone. This might require changes to the manner in which the disks are sampled, as most popular graphics libraries, including OpenGL, allow colours to be specified only at vertices, and so an accurate representation should place vertices as closely as possible to ring atoms.

To make the "pure ribbon" form of the Twister algorithm more aesthetically appealing, the endpoints of separate ribbons should join seamlessly. This might be accomplished using "blobby" object techniques, with the Marching Cubes algorithm generating the surfaces at the joins, although this would complicate the assignment of colour and the texture parameterisation at these points and in the regions surrounding them.

The PaperChain algorithm does not visualise all useful information about a carbohydrate, such as information about the side chains extending from the rings. While the ball-and-stick overlay will depict these chains, there is space for work on an algorithm that visualises information not included in PaperChain in a novel way. This is not as much of a concern for the Twister algorithm, whose primary function is to provide high-level, tertiary information.

Having established the merit of the novel algorithms, one of the main concerns is an improvement in efficiency. Rigorous analysis of the relative strengths and weaknesses of the search techniques involved is necessary to improve execution times and space costs. Additional improvements can be made in the generation of the geometry, such as generation of triangle strips and fans, which should accelerate rendering performance.

7. REFERENCES

[1] E. Angel. *Interactive Computer Graphics: A Top-Down Approach Using OpenGL*. Pearson Education, 3 edition, 2003.

[2] F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. E. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi. The Protein Data Bank: A Computer-based Archival File for Macromolecular Structures. *Journal of Molecular Biology*, 112(3):535–542, 1977. `http://www.rcsb.org/pdb`.

[3] I. Biederman and G. Ju. Surface versus Edge-Based Determinants of Visual Recognition. *Cognitive Psychology*, 20:38–64, 1988.

[4] J. Buchanan and M. Sousa. The edge buffer: A data structure for easy silhouette rendering. *Proceedings of the 1st International Symposium on Non-Photorealistic Animation and Rendering*, pages 39–42, 2000.

[5] M. Carson. Ribbon models of macromolecules. *Journal of Molecular Graphics*, 5(2):103–106, 1987.

[6] M. Carson and C. Bugg. Algorithm for ribbon models of proteins. *Journal of Molecular Graphics*, 4(2):121–122, 1986.

[7] A. Glassner, editor. *Graphics Gems*, volume 1, pages 567–571. Academic Press, 1990.

[8] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A Non-Photorealistic Lighting Model for Automatic Technical Illustration. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 447–452, 1998.

[9] W. Humphrey, A. Dalke, and K. Schulten. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.

[10] IUPAC-IUBMB Joint Commission on Biochemical Nomenclature. Nomenclature of Carbohydrates. *Pure Applied Chemistry*, 68:1919–2008, 1996.

[11] R. Raskar and M. Cohen. Image Precision Silhouette Edges. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics*, pages 135–140, 1999.

[12] J. S. Richardson. The Anatomy and Taxonomy of Protein Structure. *Advances in Protein Chemistry*, 34:167–218, 1981.

[13] H. Senay and E. Ignatius. Rules and principles in scientific data visualization. Technical report, George Washington University, 1990.

[14] E. E. Sutherland, R. F. Sproull, and R. A. Schumacker. A Characterization of Ten Hidden-Surface Algorithms. *ACM Computing Surveys*, 6(1):1–55, 1974.

[15] J. van der Zijp. FOX Toolkit. World Wide Web. `http://www.fox-toolkit.org/`.

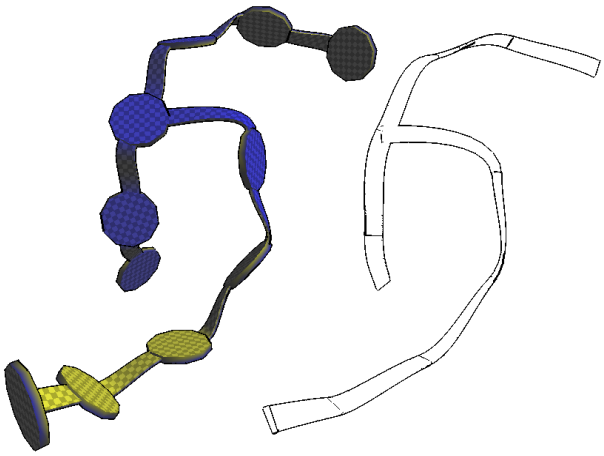

Figure 5: The Twister algorithm, illustrated on a larger molecule. The right-hand image demonstrates both the "silhouette-only" mode and the "pure ribbon" form of Twister.

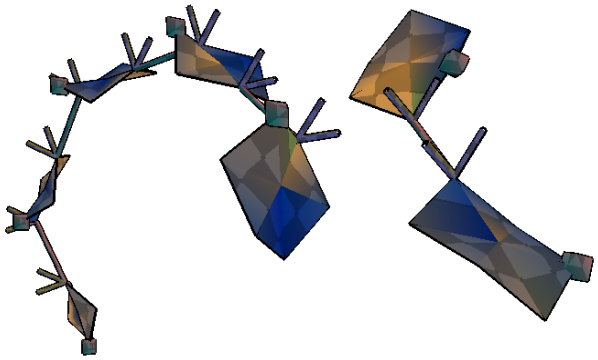

Figure 6: The PaperChain algorithm, demonstrated here on amylose (left) and celliobiose (right).

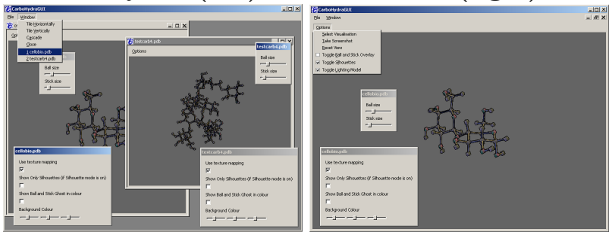

Figure 7: Two views of the interface, showing dialogues for rendering options.