Visualization of complex molecular ribbon structures at interactive rates

Conference Paper · August 2004

DOI: 10.1109/IV.2004.1320224 · Source: IEEE Xplore

Dieter W. Fellner
Fraunhofer Institute for Computer Graphics Research IGD
414 PUBLICATIONS **4,040** CITATIONS

Visualization of Complex Molecular Ribbon Structures at Interactive Rates

Andreas Halm
TU Braunschweig
Computer Graphics
halm@cg.cs.tu-bs.de

Lars Offen
TU Braunschweig
Computer Graphics
offen@cg.cs.tu-bs.de

Dieter Fellner
TU Braunschweig
Computer Graphics
fellner@cg.cs.tu-bs.de

Abstract

The increasing size of protein data available in internet databases demands new fast methods of visualization.
This paper introduces a novel visualization technique for molecular ribbon structures. The method is based on Combined BReps – a mesh structure defining a combination of polygonal faces and Catmull/Clark surfaces. Our technique provides high frame rates when interactively moving through structures of highest complexity as well as high quality images when the viewpoint remains static.

1. Introduction

The 'ribbon drawing' made popular by Richardson [15] provides an elegant method to visualize the folding and secondary structure of proteins. There are several implementations to visualize these ribbon structures, such as the one from Carson, which was improved over the years [4], [1], [2] and [3], or complete molecular visualization tools like Chimera [6], Cn3D [7], or Rasmol [14].

These implementations construct either 2D structures, e.g. ties, or only B-spline curves, or they construct a 3D ribbon with a fixed tesselation. This has the drawback that the visualization of huge ribbon structures is either unacceptably slow and of high quality, or near-interactive and of low quality.
Interactive speed is necessary for the user to get an immediate feedback while exploring the molecular structure. This increases the understanding of the visual presentation. On the other hand high quality images are needed for presentations or publications. Both – up to now conflicting – requirements can only be met by the use of 3D models supporting surface representations of variable resolution: multi-resolution meshes. There are two different popular approaches to realise such multi-resolution meshes: progressive meshes on the one hand, and subdivision surfaces on the other.
Both techniques can be used to solve the problem, but the maximum resolution of progressive meshes is bounded by the resolution of the input mesh. In contrast, the mesh defining a subdivision surface can be refined an infinite amount of times, thereby converging to the subdivision surface limit surface – a C^2-continous surface. Another advantage of the subdivision surfaces is, that they calculate only the needed data and do not, like the progressive meshes, calculate everything and then ignore most of the data.
The third more technical advantage of subdivision surfaces is that they can switch between the resolutions instantly without any additional costs, while progressive meshes need extra calculations to change between them. They are therefore better qualified for this special purpose, because the user can watch the final result without noticable delay. This is very useful for online presentations, where different views of one molecule can be shown to the audience at interactive rates.

We developed a new method to visualize these ribbon structures using a combination of simple cubic polynomial B-spline curves and a powerful boundary representation (BRep) called Combined BRep (CBR for short) [9, 11].

We will show how to construct such a CBRep from a given sequence of positions, normally the positions of the C_α atoms, to get a high quality visualization of the ribbon structure. Therefore we will give a short overview of the combined BReps in the first part of the paper. In the second part we will show how to construct the B-spline curves on which the ribbons are aligned. In part three we will construct the CBRep from the points calculated earlier in the second part.

1.1. Molecule Basics

A molecule is an accumulation of atoms, where some of them are joined by so called *bonds*. Since the ribbon structure is a visualization tool for *proteins*, we will restrict the following introduction to them. A protein is composed of possibly more than one chain of *amino acids* linked together by peptide bonds. The *backbone* of a chain is formed by three groups of atoms in each amino acid, namely the cen-

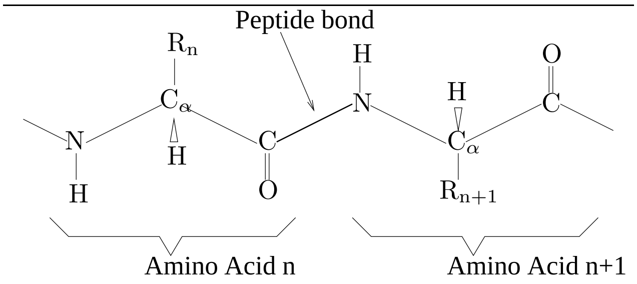

Figure 1. Structure of a protein chain: a peptide bond links amino acids in a linear way

tral atom C_α, the amino group $N-H$ and the carbonyl group $N=O$. The residue R, which is also bound to the C_α atom, charaterizes the nature of each amino acid, but is not part of the backbone. In the organic chemistry 20 amino acids are differentiated: Alanin (A), Arginin (R), Asparagin (N), Aspartat (D), Cystein (C), Glutamat (E), Glutamin (Q) Glycin (G), Histidin (H), Isoleucin (I), Leucin (L), Lysin (K), Methionin (M), Phenylalanin (F), Prolin (P), Serin (S), Threonin (T), Tryptophan (W), Tyrosin (Y) and Valanin (V).
They are usually abbreviated by using their associated character, as given in brackets above.
A protein can now be described by a sequence of these characters, e.g. IVNGEEAVPG. This is known as the *primary structure* of a protein. Since the peptide bond is not rigid, the protein can rotate around this bond to fold itself to an energetically convenient state. This state is called *tertiary structure*. This structure can be revealed by *Nuclear Magnetic Resonance (NMR)* and *X-ray crystallography* and is published for example in the *RCBS Protein Databank* [13].
Locally the tertiary structure is built by the *secondary structure* which are

- the *alpha helix*,
- the *beta sheet*
- and some random *turns*.

The secondary and tertiary structure is visualized using the ribbon drawing, which was first popularized by Richardson in [15].

Molecules consist of one or more chains, which are not connected together by atomic bonding. When drawing the ribbon structure of a molecule that contains more than one chain, one must not connect the ribbons of two chains. Chains can be handled independently of each other.

2. Combined BReps

The backbone of this ribbon implementation is the Combined BRep (CBRep), introduced by Havemann [9]. The CBRep combines a polygonal boundary representation with Catmull/Clark subdivision surfaces [5] by attaching a sharpness flag to each edge in the BRep mesh. Thereby different kinds of faces can be distinguished by the type of their vertices. We differentiate four different vertex types according to the number of sharp edges incident to them. These are *smooth*, *dart*, *crease*, and *corner* vertices, whereby a smooth vertex has no sharp edge and a corner vertex has three or more sharp edges. This leads to following face classification:

- *Smooth face*
  A face with at least one smooth edge.
- *Sharp face*
  A planar face with only sharp edges, but the vertices may be crease vertices
- *Polygonal face*
  Like a sharp face but all vertices are corner vertices, too.

According to this definition a smooth face can also have one or more sharp edges in its boundary, called *crease edges*. These edges are handled as uniform cubic B-spline curves and may create a link between smooth and polygonal areas of the mesh. A polygonal face which has a smooth face as neighbor is called a sharp face and they are connected by a common crease curve.
For each face type an appropriate tesselation and render method must be used, which can be found in detail in [9] and [11]. These methods are optimized for multi-resolution meshes, which means that the resolution can be changed without any extra effort, once the tesselation for the target resolution is calculated. Since this calculation causes additional costs, which cannot always be done in one frame, it may be distributed over more than one frame until the target resolution is reached.
The subdivision level in which a face should be rendered is controlled by the following parameters:

- view frustum clipping
- curvature
- contribution to the silhouette
- projected size
- frame rate

While the frame rate control tries to hold the triangle count below some critical value, which depends on the hardware used, the other three parameters are view dependent and set some target level which may be reached when the frame rate allows it. Through this in very complex scenes the target subdivision level may not be reached until the viewpoint is fixed, but it will ensure constant frame rates on computers with slower hardware, too. It may also happen, that neighbor faces have different levels of subdivision. To avoid cracks between those faces the tesselation of the face with

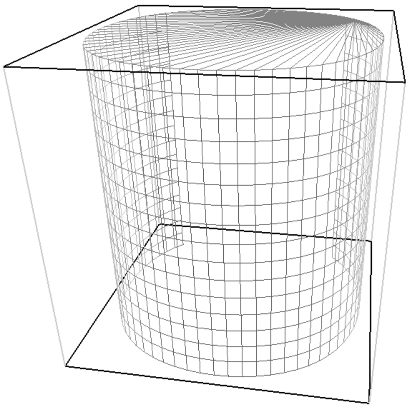

Figure 2. Generating a cylinder using CBReps

the lower resolution is automatically adapted at its boundary to match the higher level (for details we refer to [9]). This can be done with no extra effort. Only if the target resolution is finer than the tesselation calculated for this face some extra points must be evaluated.

Since the Combined BReps are based on a 2-manifold boundary representation, all operator sets generating such a mesh can be applied. In the current version a slightly modified version of the Euler-operators proposed by M. Mäntylä in [12] is used. Where each operation creating an edge has an additional boolean sharpness parameter. Therefore it is very simple to generate the needed base meshes from scratch.

Figure 2 shows a simple cylinder with only six faces in the base mesh. The black edges are sharp edges, the others are smooth. So the top and bottom are sharp faces, while the other four faces are smooth. Figure 3 extends the cylinder to a tube. Here the base mesh has fourteen faces, only one of them is a sharp face. This tube suggests how to model the complete ribbon structure with a very simple basemesh.

For this reasons the Combined BReps are an ideal tool to visualize the molecular ribbon structure. They can easily be build and provide an excellent multi resolution approach, which can switch between the different resolutions without any additional costs.

3. Cubic Spline Interpolation

Although the display of ribbon structures is just a visualization of the functionality of the secondary structures, it is preferable that the resulting ribbons pass through the positions of the C_α atoms. To achieve this, we generate the control points of the B-spline curve using the common cubic spline interpolation as described in [8]. For the sake of completeness we will recur it in this section.

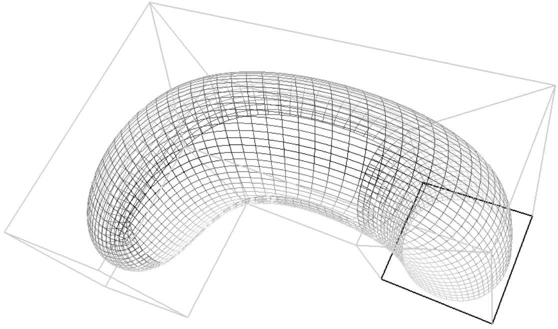

Figure 3. Generating a tube using CBReps

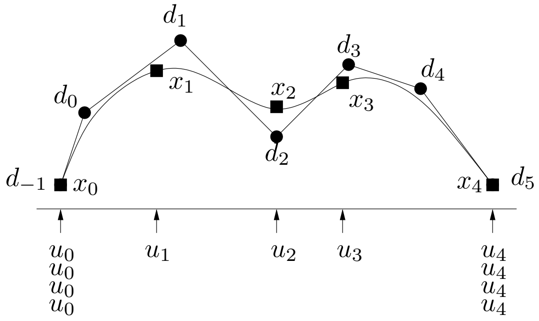

Figure 4. A B-spline curve

First we take a closer look at the relationship between data points and control points (see Figure 4 for a cubic B-spline curve).

The first step is to calculate the parameter values $u_0, \ldots, u_n$. A simple way is to set $u_i = i$ but this parameters will ignore the relative distance of the data points. So a better way is to take these distances into account. In our implementation we choose a *centripetal* parametrization. This means:

$$\begin{aligned} u_0 &= 0 \\ u_{i+1} &= u_i + \sqrt{||x_{i+1} - x_i||} \end{aligned}$$

Now we must determine the control points $d_{-1}, \ldots, d_{n+1}$ so that for the curve s holds: $s(u_i) = x_i$, so that s interpolates the data points. This means we need a linear system, which can be solved. Therefore we must recall that every B-spline curve can be written as a piecewise Bézier curve (Figure 5). In that form we have $x_i = b_{3i}$

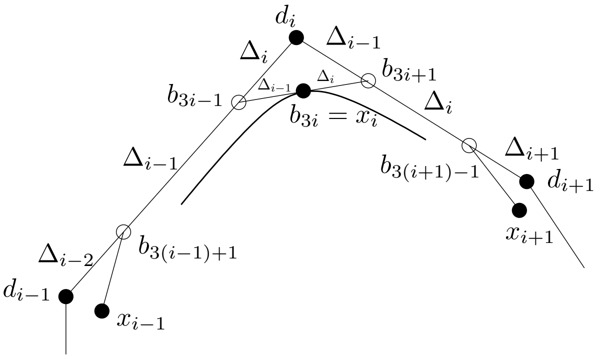

Figure 5. A Bézier segment of a B-spline curve

with $i = 0, \ldots, n$ and the inner Bézier points are related to the x_i by the first step of the de Boor algorithm as follows:

$$x_i = \frac{\Delta_i b_{3i-1} + \Delta_{i-1} b_{3i+1}}{\Delta_{i-1} + \Delta_i} \quad i = 1, \ldots, n-1$$

with $\Delta_i = u_{i+1} - u_i$. The $b_{i\pm1}$ are related to the control points d_i by:

$$\begin{aligned} b_{3i-1} &= \frac{\Delta_i d_{i-1} + (\Delta_{i-2} + \Delta_{i-1}) d_i}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i}, \\ i &= 2,\ldots,n-1 \\ b_{3i+1} &= \frac{(\Delta_i + \Delta_{i+1}) d_i + \Delta_{i-1} d_{i+1}}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}}, \\ i &= 1,\ldots,n-2 \end{aligned}$$

Near the ends we have some special cases:

$$\begin{aligned} b_2 &= \frac{\Delta_1 d_0 + \Delta_0 d_1}{\Delta_0 + \Delta_1} \\ b_{3n-2} &= \frac{\Delta_{n-1} d_{n-2} + \Delta_{n-2} d_{n-1}}{\Delta_{n-2} + \Delta_{n-1}} \end{aligned}$$

Now we can eliminate the b_i and get:

$$(\Delta_{i-1} + \Delta_i) x_i = \alpha_i d_{i-1} + \beta_i d_i + \gamma_i d_{i+1}$$

with $\Delta_{-1} = \Delta_n = 0$ and:

$$\begin{aligned} \alpha_i &= \frac{\Delta_i^2}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i} \\ \beta_i &= \frac{\Delta_i(\Delta_{i-2} + \Delta_{i-1})}{\Delta_{i-2} + \Delta_{i-1} + \Delta_i} + \frac{\Delta_{i-1}(\Delta_i + \Delta_{i+1})}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}} \\ \gamma_i &= \frac{\Delta_{i-1}^2}{\Delta_{i-1} + \Delta_i + \Delta_{i+1}} \end{aligned}$$

After choosing b_1 and b_{3n-1} arbitrarily, we eventually obtain the following linear system:

$$\begin{bmatrix} 1 & & & & \\ \alpha_1 & \beta_1 & \gamma_1 & & \\ & & \ddots & & \\ & & \alpha_{n-1} & \beta_{n-1} & \gamma_{n-1} \\ & & & & 1 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix} = \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{n-1} \\ r_n \end{bmatrix}$$

With $r_0 = b_1$, $r_i = (\Delta_{i-1} + \Delta_i) x_i$ and $r_n = b_{3n-1}$. The first and the last polygon vertices are $d_{-1} = x_0$ and $d_{n+1} = x_n$. Due to the simple band structure of the matrix this linear system can be solved by LU-decomposition. A more detailed description can be found in [8].

4. Generating Ribbons

The *Brookhaven Protein Databank format (pdb)* delivers information about the secondary structure of the molecule. This structure may consist of a number of chains which do not interfere with each other, with respect to display ribbon structures. They can be handled separately, and the following steps can be done once for each chain from a given molecule. The file defines a sequence of amino acids which forms either a helix, a sheet or a turn. Each of these acids has a C_α atom through which the resulting B-spline curve should pass. As mentioned above, not all C_α atoms are part of a secondary structure. These atoms fill the gap between two given sequences and will be handled like they belong to a turn for each such gap. At the junction between two such sequences we must generate an additional data point, because one sequence ends at one residue and the other starts at the next one. To close this gap the additional point must be calculated. Two additional coordinates are needed, one at each end of the chain. These coordinates will ensure that all C_α atoms lie inside the generated ribbons, otherwise the two C_α atoms at the ends of the chain would lie outside of the ribbons. As the display of the ribbons is just a visual aid for understanding a molecule's function, it is justifiable to make an educated guess for these positions.

To get a 3-dimensional ribbon from the positions of the C_α atoms we must generate some 3D vectors which span the ribbon structure. For every amino acid A_i in the chain of length n with its C_α positioned at p_{A_i}, the ribbon height vector h_{A_i} is

$$h_{A_i} = \frac{p_{A_i} - \frac{p_{A_{i-1}} + p_{A_{i+1}}}{2}}{\|p_{A_i} - \frac{p_{A_{i-1}} + p_{A_{i+1}}}{2}\|}.$$

This is always defined because no three adjacent C_α positions are collinear. The ribbon width vector w_{A_i} is

$$w_{A_i} = \frac{h_{A_i} \times (p_{A_{i+1}} - p_{A_i})}{\|h_{A_i} \times (p_{A_{i+1}} - p_{A_i})\|}.$$

To avoid twisted ribbons, the resulting vectors h_{A_i}, w_{A_i} are compared to $h_{A_{i-1}}$ and may be corrected by rotation and exchanging, so that the dot product $\langle h_{A_{i-1}}, h_{A_i}\rangle \geq 0$ and $\langle h_{A_{i-1}}, h_{A_i}\rangle \geq \langle h_{A_{i-1}}, w_{A_i}\rangle$. This problem was mentioned by Carson in [4] as carbonyl oxygen flip.

For the residues at the two ends of a chain, the formulae are slightly modified. At the beginning of a chain A_{i-1} does not exist, so

$$h_{A_0} = \frac{(p_{A_1} - p_{A_2}) \times (p_{A_1} - p_{A_0})}{\|(p_{A_1} - p_{A_2}) \times (p_{A_1} - p_{A_0})\|}$$

and

$$w_{A_0} = \frac{h_{A_0} \times (p_{A_1} - p_{A_0})}{\|h_{A_0} \times (p_{A_1} - p_{A_0})\|}.$$

Likewise, at the end of a chain A_n does not exist, and

$$h_{A_{n-1}} = \frac{(p_{A_{n-3}} - p_{A_{n-2}}) \times (p_{A_{n-1}} - p_{A_{n-2}})}{\|(p_{A_{n-3}} - p_{A_{n-2}}) \times (p_{A_{n-1}} - p_{A_{n-2}})\|}$$

$$w_{A_{n-1}} = \frac{h_{A_{n-1}} \times (p_{A_{n-1}} - p_{A_{n-2}})}{\|h_{A_{n-1}} \times (p_{A_{n-1}} - p_{A_{n-2}})\|}.$$

The vector correction based on the dot product, mentioned above, is made for the residue A_{n-1}, too.

The two additional points at the end of the chain can now be calculated as:

$$\begin{aligned} p_{A_{-1}} &= c\,(h_{A_0} \times w_{A_0}) \\ p_{A_n} &= -c\,(h_{A_{n-1}} \times w_{A_{n-1}}) \end{aligned}$$

The constant c is chosen empirically as $c = 1.7$. This results in nice ends of the ribbons. Other choices of c are possible, but may result in the first or last C_α atom lying outside of the generated ribbon structure. The list of the C_α positions p_{A_i} including the two points $p_{A_{-1}}$ and p_{A_n}, is called the *base list*. The coordinate positions given by the base list are used to produce a cubic B-spline polygon of an interpolating C^2 cubic spline curve, as described in Section 3.

The resulting cubic B-spline polygon with the parameter values $u_0, \ldots, u_{n-1}$ is then used to insert additional data points into the base list. Therefore the B-spline curve is evaluated at the joint of two sequences $s_0 = \{p_{A_i}, \ldots, p_{A_{i+j}}\}$ and $s_1 = \{p_{A_{i+j+1}}, \ldots, p_{A_{i+j+1+k}}\}$ at the point $(u_{i+j} + u_{i+j+1})/2$. If s_0 is a sheet, we will also evaluate the B-spline at $(u_{i+j-1} + u_{i+j})/2$. This is necessary to give us the needed auxiliary points p_{H_j} for the transition between two sequences (*transition point*) and for generating the arrows (*arrow point*). If a residue based coloring is used, additional transition points must be calculated between every amino acids in the sequence. For each auxiliary point the vectors $h_{p_{H_j}}$ and $w_{p_{H_j}}$ are calculated as above with the proper $p_{A_i}, p_{A_{i+1}}$ as neighbors. After relabeling we get m points p_i with the corresponding h_{p_i} and w_{p_i}.

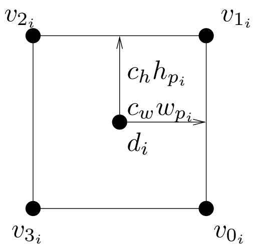

Figure 6. At every control point d_i one quad is generated, as the cross sections of the ribbon control mesh. The scale factors c_h, c_w depend on the type of secondary structure. The vectors h_i and w_i span the normal plane at d_i, ie. the plane that is normal to the tangent of the Bezier curve at d_i.

After this we restart the spline interpolation to get a B-spline curve passing throung all the data points including the auxiliary points. The resulting control vertices $d_0, \ldots, d_{m-1}$ of the B-spline curve are used to generate the basemesh of the ribbon structure.

For each d_i we generate a quad q_i as seen in figure 6. Now the generation of the complete basemesh is very simple. As mentioned above we can use standard Euler-operations to create the base mesh. For this purpose we take two quads q_i and q_{i+1} with the corners v_0, v_1, v_2, v_3 and w_0, w_1, w_2, w_3. If $i = 0$, we generate a face with only smooth edges with the corners of q_0, otherwise we have already generated the face associated with q_i. Then we are connecting v_0 to w_0, v_1 to w_1 and so forth with smooth edges. At last we are building q_{i+1} with smooth edges. Figure 7 illustrates this procedure. If q_{i+1} is not the last quad, we start over. At the q_i, which are generated by a transition point, we can for example change the color of the basemesh according to the secondary structure or even at residue level.

4.1. Generating arrows in the ribbon

The generation of arrows in the structure to visualize a direction is very simple through the usage of CBReps. They can build by just adapting the scaling factor of one normal plane q_i. By just changing one factor you will get a flat arrow like in Figure 9, which can be used for the sheets, or by changing both factors a round arrow can be achieved for the helices.

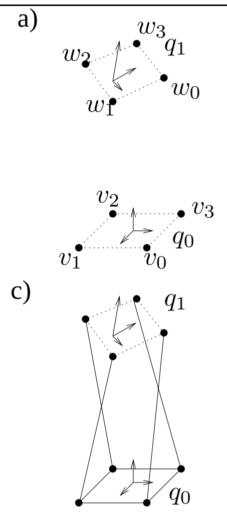

Figure 7. Building one element of the tube

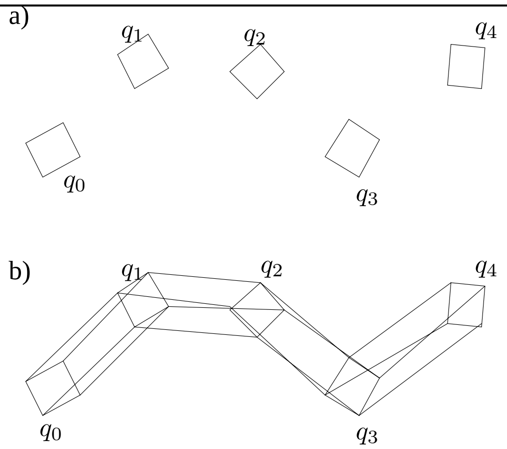

Figure 8. Building the tube from the q_i

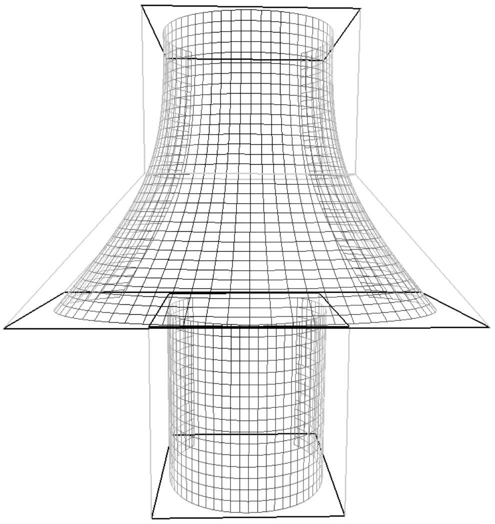

Figure 9. Generating an arrow using CBReps

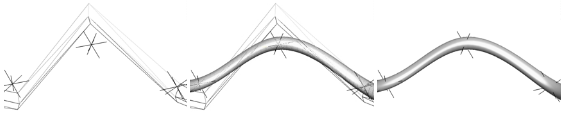

Figure 10. Computation of the base mesh. The stars show the positions of the C_α atoms. While the generated base mesh does not need to include the C_α positions, the spline curve that is generated at run time always does.

4.2. The resulting basemesh

The generated basemesh is very simple, compared to the visual quality that is achieved through subdivision. Each sequence of n residues results in: (without arrows)

- $f = 4(n+1)+2$ faces
- $e = 4(2n+3)$ edges
- $v = 4(n+2)$ vertices

From these vertices only eight are not *regular*, which means they have a degree not equal to four. Since these irregularities are very regional they will not disturb the visual impression. All faces (except the two faces per arrow which are sharp faces) are smooth, regular Catmull/Clark faces of degree four. Each of these faces results in $(2^n+1)(2^n+1)$ quads each in resolution depth n, e.g. $16 \cdot 16$ quads, resulting in 324 vertices, in depth 3 for each face in this resolution. This leads to a very high quality of the images produced. Since the quality of the faces depends on the projected size, too, only faces close to the spectator can reach the highest resolution. Therefore the resulting data is still small enough for even older graphic boards.

5. Results

The described algorithm is capable of generating images with very high detail. Additionally, setting the visible detail dynamically based on the factors described in Section 2 allows us to reach significantly better framerates on slower computers. Table 1 shows the resulting framerates for some molecules. The whole molecule was always visible, so no

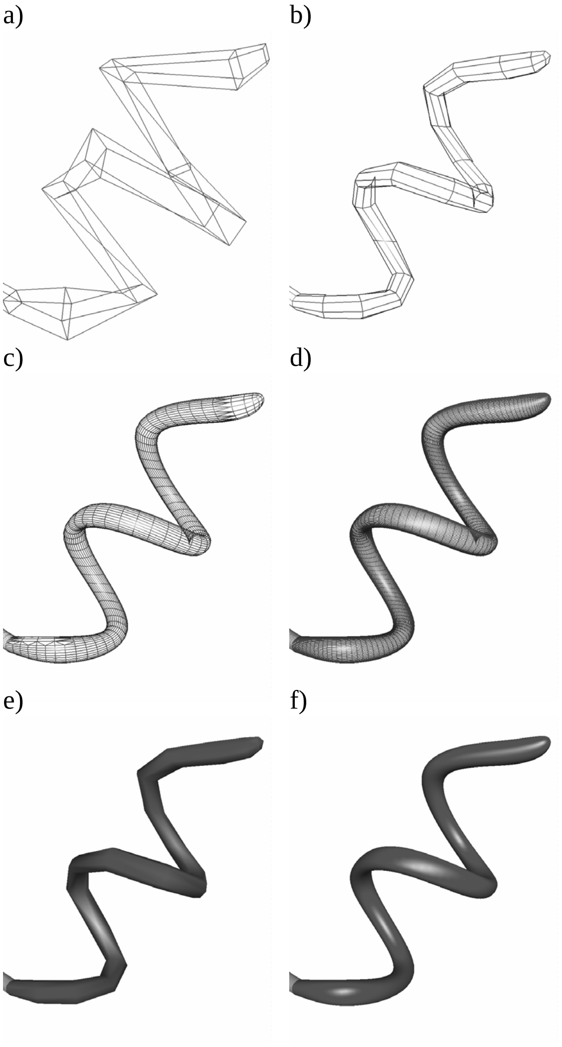

Figure 11. Ribbon computation for a helix structure. a) The generated base mesh. b) Very low detail wireframe display of the ribbon mesh. c) Medium detail wireframe display. d) High detail wireframe display. e) Very low detail display of the ribbon mesh. f) High detail display of the ribbon mesh. While moving the structure, the detail is automatically adjusted to reach a framerate of 20–24 fps. For still images, the highest detail setting is used.

view frustum culling took place. All the tests were made on an Athlon 1200 with a GeForce 5200 graphics card.

The column *High detail* lists the framerate we got when always running the highest detail setting possible. The column *Adaptive detail* lists the results we got with an adaptive method that switches the available levels of detail dependend on the current scene.

Conclusions

In this paper we have presented a very powerful visualization method for molecular ribbon structures. It is based on the so called CBReps, a combination of polygonal and subdivision surfaces. This method has been combined with the procedure of spline interpolation to generate nice ribbons that pass through the C_α positions. Using Combined BReps together with a sophisticated method to generate the base mesh is very simple, and can easily be used for other purposes.

As can be seen by the example images (Figure 12), the rendering quality is very high for still images. The approach also allows the use of advanced level-of-detail algorithms. As shown in Figure 11b, the generated mesh can also be very low detailed, which allows interactive work with complex molecules on slower computers. That low-detail mesh is sufficiently detailed to recognize the structure of the molecule. The quality of that approach can also be seen by comparing the rows *High detail* and *Adaptive detail* of Table 1.

It is planned to integrate forms of hardware acceleration for the display of the ribbon mesh. OpenGL extensions like *Compiled Vertex Arrays* and *Vertex Buffer Objects* come into mind there. Likewise, it is possible to use *Multisample Buffers* to generate antialiased images of even higher quality.

The functionality and performance of current *Vertex Shaders* and *Pixel Shaders* does not seem to be sufficient to use them for generating the ribbon structure. That might be of interest in the future though.

Currently, the ribbons are colored by their type. It is also possible to color them differently, for example according to their amino sequence.

6. Acknowledgement

The funding of this project by the German Research Foundation (DFG) under grant Fe 431/6-1 is gratefully acknowledged. Further credit goes to Sven Havemann for his Compbined Breps and to our partners in the project from the structural biology side Dirk Heinz, Guido Dieterich, and Joachim Reichelt.

				Secondary		Frames per second	
Name	PDB ID	Atoms	Amino Acids	Structures	Chains	High detail	Adaptive detail
HIV-1 Capsid	1A43	573	72	13	1	29	30
Mutant of Carp Parvalbumin	1B8L	806	108	13	1	21	22
Oxid. Dsba Crystal Form II	1A2J	1449	188	21	1	12	24
Hemoglobin	1IDR	1867	253	33	2	9	24
Ribonuclease	11BA	1925	248	38	2	9	24
Antigen-Antibody Complex	2HRP	6852	875	178	6	3	25
Ribosomal Subunit	1FJG	51770	3897	346	22	1	10
Ribosomal Subunit	1FFK	64268	6484	29	29	1	7

Table 1. Display speed of the generated ribbons (with interactivity constraint disabled)

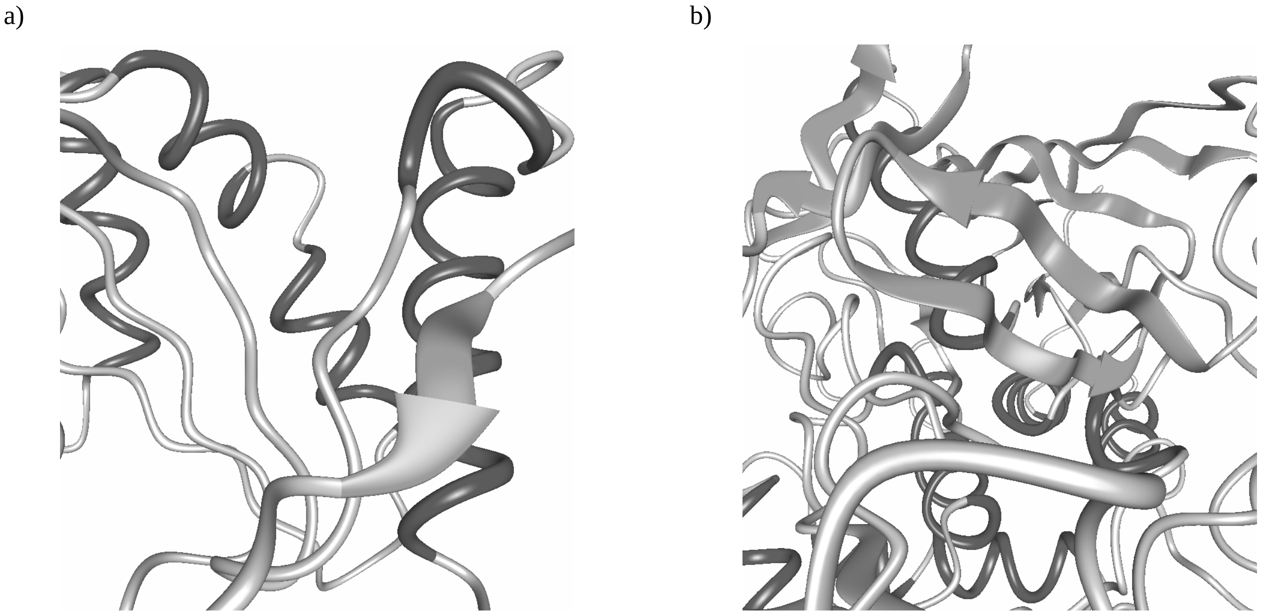

Figure 12. Result images for molecules, rendered with high detail. a) Oxidized Dsba Crystal Form II (21 ribbons). Note the arrows generated in the sheets. b) Cholesterol Oxidase (69 ribbons)

References

[1] M. Carson. Ribbon models of macromolecules. *J.Mol.Graphics*, pages 103–106, 1987.

[2] M. Carson. Ribbons 2.0. *J.Appl.Cryst.*, pages 958–961, 1991.

[3] M. Carson. Ribbons. *Methods in Enzymology*, pages 493–505, 1997.

[4] M. Carson and C. Bugg. Algorithm for ribbon models of proteins. *J.Mol.Graphics*, pages 121–122, 1986.

[5] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, pages 350–355, September 1978.

[6] Chimera. http://www.cgl.ucsf.edu/chimera/.

[7] Cn3D. ftp://ftp.ncbi.nih.gov/cn3d/.

[8] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, Inc., 1990.

[9] S. Havemann. Interactive rendering of catmull/clark surfaces with crease edges. *The Visual Computer*, pages 286–298, 2002.

[10] S. Havemann and D. Fellner. Generative mesh modeling. Technical Report TUBSCG-2003-01, Institute of Computer Graphics, TU Braunschweig, 2003.

[11] S. Havemann and D. Fellner. Progressive combined breps - multi-resolution meshes for incremental real-time shape manipulation. Technical Report TUBSCG-2003-05, Institute of Computer Graphics, TU Braunschweig, 2003.

[12] M. Mäntylä. *An Introduction to Solid Modeling*. Computer Science Press, Rockville, 1988.

[13] RCBS protein database. http://www.pdb.org.

[14] Rasmol. http://www.umass.edu/microbio/rasmol/.

[15] J. Richardson. The anatomy and taxonomy of protein structure. *Adv. Protein Chem.*, pages 167–339, 1981.