EUROGRAPHICS - IEEE VGTC Symposium on Visualization (2005)
K. W. Brodlie, D. J. Duke, K. I. Joy (Editors)

BioBrowser: A Framework for Fast Protein Visualization

Andreas Halm[†] and Lars Offen [‡] and Dieter Fellner[§]

Institut für ComputerGraphik, TU Braunschweig, Germany

Abstract

This paper presents a protein visualization system called BioBrowser, *which provides high quality images at interactive frame rates for molecules of extreme size and complexity. This is achieved by a shift in the tessellation approach: triangle meshes are not produced a priori on a 'just-in-case' basis. Instead, tessellation happens 'just-in-time' given a certain camera position, image size and interaction demand. Thus, our approach is based on multi-resolution meshes and on new extensions of graphics hardware. The paper shows how to reduce geometric data by using subdivision surfaces for ribbon structures and molecular surfaces and by using billboards instead of spheres consisting of triangles. It also shows how to use fragment shaders to create a three dimensional appearance and realistic sphere intersections. The combination of these approaches leads to an image quality not yet seen in interactive visualization environments for molecules of that size/complexity. All the above methods are combined to gain a high performance configurable visualization system on standard hardware.*

Categories and Subject Descriptors (according to ACM CCS): H.4.3 [Information System Applications]: Communications Applications – Information browsers I.3.8 [Computing Methodologies]: Computer Graphics – Applications J.3 [Computer Applications]: Life and Medical Sciences – Biology and genetics;

1. Introduction

Protein visualization at interactive rates has become more and more important, as the number of proteins in the *RCBS Protein Data bank* [pdb] is increasing very fast. The ability to visualize the 3D structure of these proteins is critical in various areas such as drug design or protein modeling, because the function, which means the possible interactions with other molecules, of a protein is closely connected to its 3D structure. The prediction of the 3D structure of a protein is for example addressed in [KHM∗03]. Therefore fast visualization tools are required, which can handle proteins with a huge number of atoms.

Interactive rates are even more important if 3D molecular structures are being used to store, locate, and access information (in any textual or multimedia format) about the respective molecule.

Of course, there are visualization tools like Chimera [Chi], Cn3D [cn3], Rasmol [ras], or FPV [CWWS03] but they either exhibit a significant drop in rendering performance when handling very large proteins or have limited visualization styles for, e.g., ribbon structures or space fill.

In this paper we present the visualization components of a new tool called *BioBrowser* which aims at supporting bio scientists in the information access in a similar way web browsers support users accessing material from the web. The visualization components currently provide fast and accurate visualization routines for the following common styles:

- Balls and Sticks
- Sticks
- Space-fill
- Ribbons
- Surfaces

Due to the modular architecture of the *BioBrowser* it is easy to attach various other visualization styles, but so far we have limited them to the most common ones. That is to say we haven't invented new visualization styles yet, but we applied

† e-mail: a.halm@cg.cs.tu-bs.de
‡ e-mail:l.offen@tu-bs.de
§ e-mail:d.fellner@tu-bs.de

various methods of modern computer graphics, like multi-resolution methods, as well as extensions of modern graphic boards like fragment shaders, to achieve interactive frame rates while preserving a high visual quality in the resulting renderings. These methods are described in detail in Section 3.

1.1. Molecule Basics

A molecule is an accumulation of atoms, where some of them are joined by so called *bonds*. Since the BioBrowser is in the first place a visualization tool for *proteins*, we will restrict the following introduction to this domain. A protein is composed of possibly more than one chain of *amino acids* linked together by peptide bonds. The *backbone* of a chain is

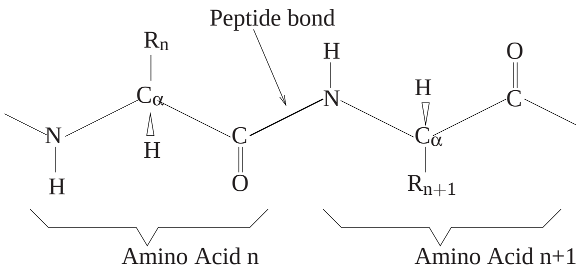

Figure 1: *Structure of a protein chain: a peptide bond links amino acids in a linear way*

formed by three groups of atoms in each amino acid, namely the central 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, characterizes the nature of each amino acid, but is not part of the backbone.
A protein can be described by a sequence of abbreviations for the different amino acids, e.g. by '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 Data bank* [pdb].

Locally, the tertiary structure is built by the *secondary structure* which consists of *alpha helices*, the *beta sheets* and *turns*. The secondary and tertiary structure can be visualized using the ribbon drawing, introduced by Richardson in [Ric81]. Molecules may 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.

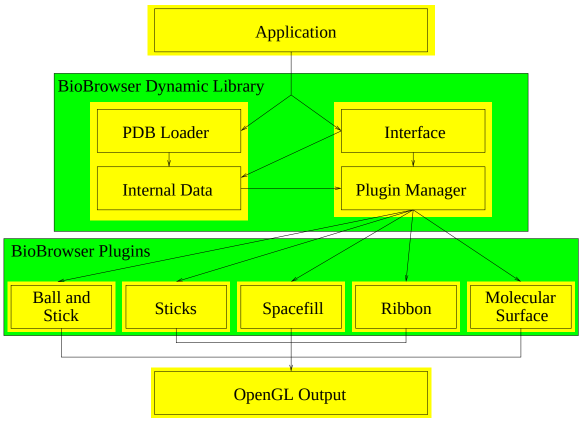

Figure 2: *The interconnection between the* BioBrowser *modules. The five display modules can be used independently, even in the same frame, to blend visualization styles.*

2. System Overview

The *BioBrowser* consists of several modules (see Figure 2). The core modules are the visualization modules, each of which implements one visualization style. The visualization modules can be used independently of each other but they can also be mixed in a single frame using alpha blending. This will allow fairly advanced molecule display when used together with additional information about the displayed molecule. Additionally, a PDB file loader module is implemented to allow easy generation for the internal data structures. All these modules together form a dynamic library, which is currently used with the *BioBrowser* executable, but can also be used to allow molecule display embedded in other applications, for example Internet browsers or CAVE®-applications.

3. Visualization Concepts

The general concept of all visualization modules is to reduce/avoid geometry data whenever possible. Consequently, we replace polygonal meshes by *billboards* and ordinary meshes by multi-resolution meshes,

3.1. Ball-and-Stick Model

The Ball-and-Stick visualization style shows all existing bonds in the molecule as well as all atoms as equal-sized spheres (Figure 4). Since the spheres will not intersect each other it is not necessary to model the spheres as a polytope. Instead, it is sufficient to model an atom as just a quadrangle together with a texture resembling a sphere, called a *billboard*, whose normal will always be directed towards the viewer (Figure 3 (a)). The bonds are realized in an analogous manner (Figure 3 (b)). To create the illusion of 3-dimensional objects the bonds do not connect the mid points of the atoms, but are a bit shorter, so that the impression arises that the bonds intersect with the atoms. The use of

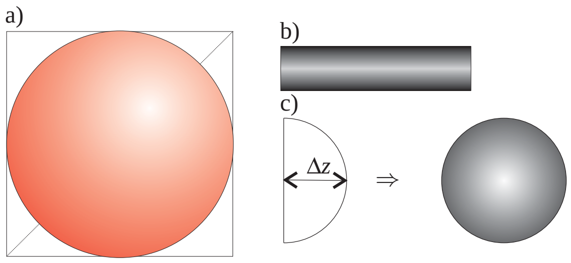

Figure 3: *Billboard for an atom (a) and a bond (b), Generating the alpha channel of the 2D texture (c)*

billboards reduces the amount of geometric data by orders of magnitude. The drawback is that the billboards must be rotated and translated, so that the normal points to the spectator, instead of just translating the spheres. The calculation of this transformation matrix is slightly more expensive than to set up the translation matrix, but afterwards the graphic hardware must only transform the four vertices of the billboard and not every vertex of the sphere. Therefore the drawback changes to a gain when comparing the billboard to spheres of modest to high resolution.
Although some of the perspective information is lost, in this case the difference is hardly noticeable.

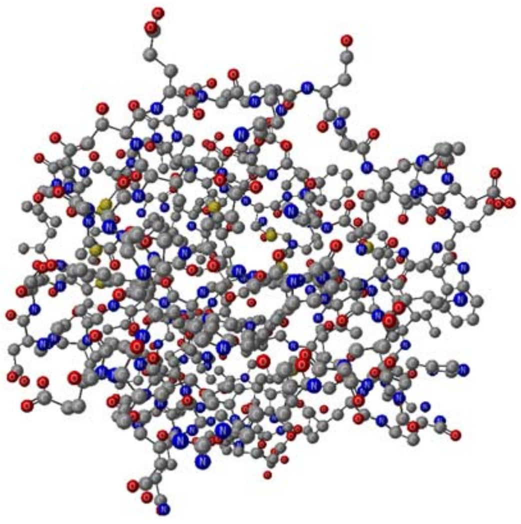

Figure 4: *Ball-and-Stick visualization of protein* 1axq

3.2. Space-fill (van der Waals) Model

To give an overall impression of the tertiary structure the protein is visualized with spheres having the *van der Waals* radius of the related atom. Figure 5 demonstrates this style, called space-fill.

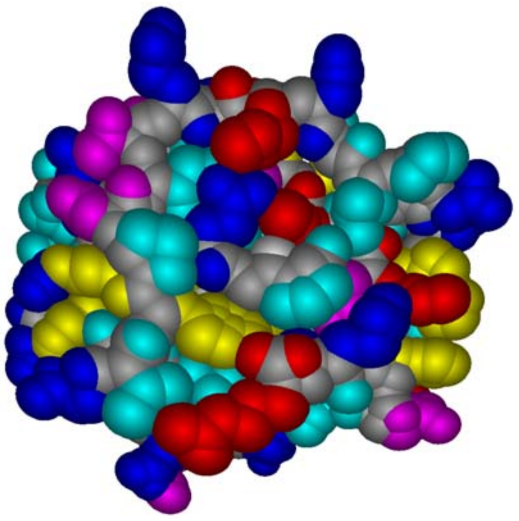

Figure 5: *Space-fill visualization of protein* 1axq

In this display mode the spheres do intersect each other and therefore billboards cannot be used as they won't intersect properly. Thus, a 2D texture with alpha channel is used. The alpha channel models the relative z-depth of the unit half sphere (Fig. 3 b)). These 'enhanced' billboards can be rendered by using a fragment program which translates the pixels of the billboard according to the value in the alpha channel (see Figure 6). As shown in the figure the

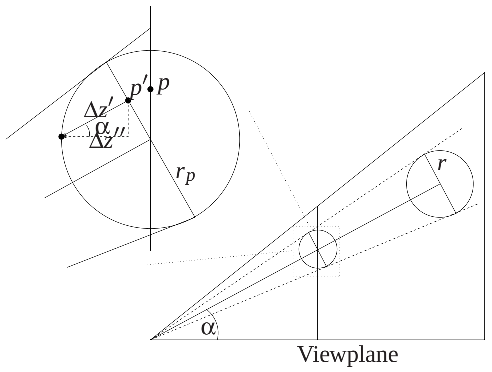

Figure 6: *How to correct the saved z-depth of the billboard when projecting it onto the viewplane.*

relative z-depth of the billboard must be corrected by the projective radius of the sphere r_p and the factor $\cos\alpha$, so that $\Delta z'' = \Delta z' * \cos\alpha = r_p * \Delta z * \cos\alpha$ holds. The correction factor $r_p * \cos\alpha$ is a simplification, because we assume a parallel projection of an atom after having projected it to the viewplane. This increases the rendering speed and in normal viewing perspective it won't disturb the visual impression.
The resulting fragment program is written as both

`GL_ARB_fragment_program` extension and `GL_NV_fragment_program` extension. This is done to allow specific optimizations for NVidia hardware. The fragment program takes two parameters: the 2D texture and the correction factor $r_p * \cos\alpha$ as a vertex attribute. The texture is evaluated per pixel and while the red, green and blue values are multiplied with the current primary color to get the effect of the `GL_MODULATE` texture environment, the alpha value of the texture is multiplied with the correction factor (given as a parameter) and added to the fragment's z-coordinate. As per-pixel depth testing takes place after the fragment program has been executed, intersection between objects are correctly rendered. This method is also known as *depth sprites*. A more detailed description can be found in [SGwHS98].

For older graphic hardware which does not support fragment programs or which have only slow implementations of the fragment programs the *BioBrowser* implements a fallback solution. Several precalculated half-spheres with different resolutions are currently used. The correct level of detail is chosen by the projective size of the sphere on the viewplane. Then the half-spheres are rendered instead of the billboards.

3.3. Bond Model

In this model only the bonds of the molecule are drawn. The bonds are realized as precalculated meshes of half-cylinders which are rotated like the billboards. To close the gaps between such half-cylinders precomputed half-spheres are used.

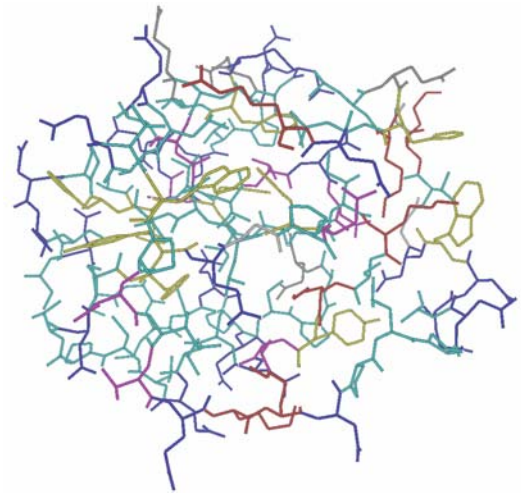

Figure 7: *Stick visualization of protein* 1axq

3.4. Ribbon Model

The 'ribbon drawing' introduced by Richardson [Ric81] provides an elegant method to visualize the folding and secondary structure of proteins. There are several implementations to visualize these ribbon structures like the program from Carson which was improved over the years [CB86], [Car87], [Car91] and [Car97], or complete molecular visualization tools like Chimera [Chi], Cn3D [cn3], or Rasmol [ras].
These implementations construct either 2D structures, e.g., B-spline curves, or they construct a 3D ribbon with a fixed tessellation. This has the drawback that the visualization of huge ribbon structures is either unacceptably slow but of acceptable quality, or it is interactive but of low quality – particularly when viewed at closer distance.

Interactive speed is necessary for the user to get an immediate feedback while exploring the molecular structure. 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: multiresolution meshes.

We developed a new method ([HOF04]) to visualize these ribbon structures using a combination of simple cubic polynomial B-spline curves, which build the skeleton of the resulting ribbons and a powerful boundary representation (BRep) called *Combined BReps* (CBRep for short) [Hav02, HF03b]. The CBRep combines a polygonal boundary representation with Catmull/Clark subdivision surfaces [CC78] by attaching a sharpness flag to each edge in the BRep mesh. The control points of a B-spline curve defining the skeleton of the ribbons can be determined by an interpolation process ([Far90]) so that the resulting curve will pass through the positions of the C_α atoms.

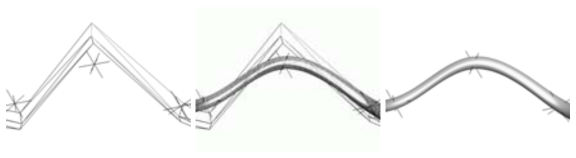

Figure 8: *Computation of the base mesh. The stars show the positions of the C_α atoms. While the generated base mesh does not need to include the C_α positions, the spline curve that is generated at run time always does.*

By using the CBRep structure it is now quite easy to generate the ribbon structures: each control vertex is extended to a quad (representing the cross section) and two quads are joined to an element of the ribbon (see Figure 8). The method used is able to generate images with very high detail. Additionally, setting the visible detail dynamically according to the parameters

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

allows us to reach interactive frame rates on even slowest computers. Figure 9 displays some tessellations for different levels of detail. Table 1 shows the resulting frame rates for some selected molecules. The whole molecule was always visible, so no view frustum culling took place. All the tests were made on an Athlon 1200 with a GeForce 5200 graphics card. It is worth noting that our approach also significantly

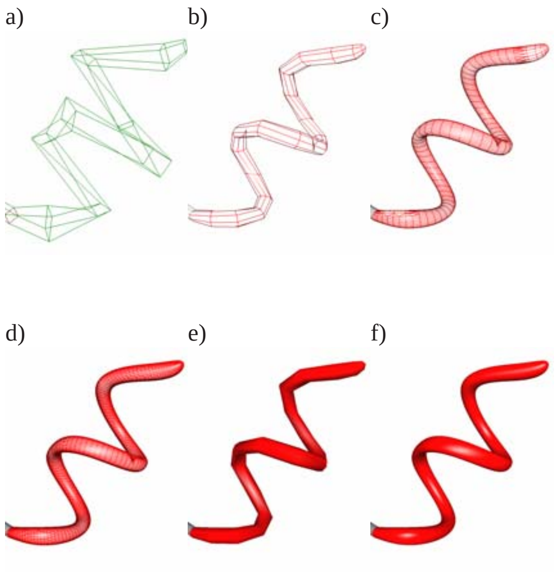

Figure 9: *Ribbon computation for a helix structure. a) The generated base mesh. b) Very low detail wireframe display of the ribbon mesh. c) Medium detail wireframe display. d) High detail wireframe display. e) Very low detail display of the ribbon mesh. f) High detail display of the ribbon mesh. While moving the structure, the detail is automatically adjusted to reach a frame rate of 20–24 fps. For still images, the highest detail setting is used.*

speeds up the culling process as a single test for the complete (in)visibility of one CBRep patch eliminates the many culling tests which were necessary if the ribbons had been tessellated during preprocessing resulting in tons of individual triangles, each of which had to be tested separately.

The columns *High detail/Low detail* give the frame rate resulting from always running the highest/lowest available detail setting. The column *Adaptive detail* lists the frame rates with our adaptive method that switches the level-of-detail according to the current environment determined by the graphics and computing hardware, the scene complexity, and the viewing position taking the parameters mentioned

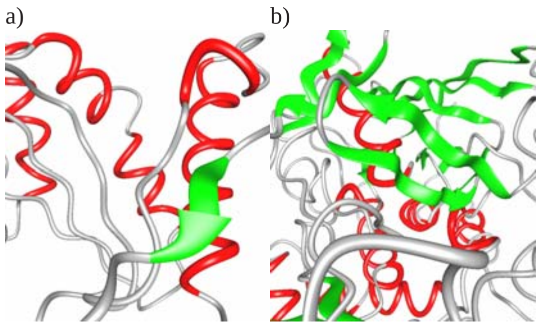

Figure 10: *Result images for molecules, rendered with high detail. a) Oxidized Dsba Crystal Form II (21 ribbons). Note the arrows generated in the sheets. b) Cholesterol Oxidase (69 ribbons)*

above into account. The number of triangles rendered in the different modes is also shown, whereas the number for adaptive level of detail is an average value.

3.5. Molecular Surface Model

In addition to the *van der Waals surface* described in Section 3.2 two other molecular surfaces can be defined by using a sphere with radius r_p, called probe, which is rolled over the van der Waals surface. The first one is the *Solvent Accessible Surface (SAS)* defined by Lee and Richard in [LR71] as the topological boundary of the molecule spheres with their radius increased by r_p.

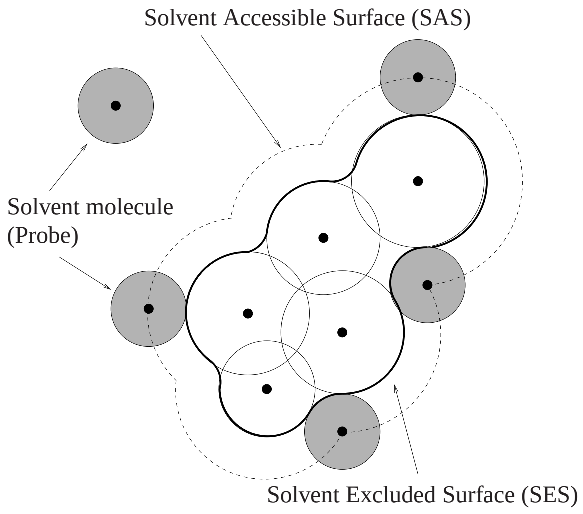
Solvent Accessible Surface (SAS)
Solvent molecule
(Probe)
Solvent Excluded Surface (SES)

Figure 11: *The solvent accessible surface (SAS) is traced by the center of the probe. The solvent excluded surface (SES) is the topological boundary of the union of all possible probes which do not overlap with the molecule.*

The other is called *Solvent Excluded Surface (SES)* after

Name	PDB ID	Atoms	Amino Acids	Secondary Structures	Chains	fps/#tris rendered High	Low	Adaptive detail
HIV-1 Capsid	1A43	573	72	13	1	29/163,000	>30/1,600	30/160,000
Calcium-Bound Triple Mutant Of Carp Parvalbumin	1B8L	806	108	13	1	21/236,500	>30/2,300	22/145,000
Oxidized Dsba Crystal Form II	1A2J	1,449	188	21	1	12/413,000	>30/4,045	24/80,000
Hemoglobin	1IDR	1,867	253	33	2	9/565,000	>30/5,520	24/33,000
Ribonuclease	11BA	1,925	248	38	2	9/580,000	>30/5,725	24/31,000
Antigen-Antibody Complex	2HRP	6,852	875	178	6	3/2.200,000	25/22,000	25/25,000
Ribosomal Subunit	1FJG	51,770	3,897	346	22	1/5.500,000	10/54,000	10/54,000
Ribosomal Subunit	1FFK	64,268	6,484	29	29	1/7.200,000	7/70,000	7/70,000

Table 1: *Display speed of the generated ribbons*

[GB78], which is the topological boundary of all possible probes having no intersection with the molecule. This surface is a refinement of the *Smooth Molecular Surface* which was defined by Richard in [Ric77]. Figure 11 shows the difference between the two surfaces in 2D.

The generation of the solvent excluded surface can be divided into four steps:

1. calculate an auxiliary surface
2. convert this auxiliary surface into an analytic surface
3. treatment of special cases
4. generate surface for visualization

The auxiliary surface needed by the algorithm is equivalent to the alpha-shape with an alpha value equal to the probe radius [EM94]. Therefore many implementations like [BLMP97] uses them to solve the first step. In our implementation we use the *reduced surfaces* established by Sanner in [San92] and [SO96]. The reduced surfaces are a fast alternative to the alpha-shape calculation as their generation can be done in $O[nlog(n)]$ where as the running time of the alpha-shape computation is $O[n^2]$ with n is the number of atoms in the molecule. Figure 12 shows the reduced surface for the protein *1a3i*. In general, we follow the algorithm proposed by Sanner but we improve the treatment of the special cases. The main difference to other applications in this area is that we are not generating a triangulated surface from the analytical surface, but converting it to a subdivision surface

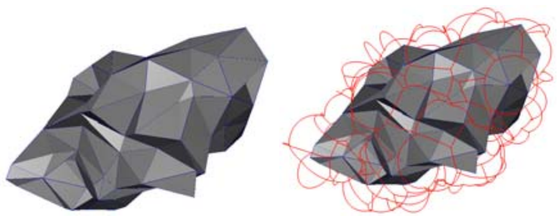

Figure 12: *Reduced surface for the protein* 1a3i. *On the right the SAS edges are shown, too.*

using Bèzier patches. Similar to the visualization modes presented so far, this allows a lazy evaluation approach producing the tessellation 'just in time' for the viewing parameters given instead of producing a tessellation 'just in case' the viewing position comes close to the surface (with the risk that the actual viewing position comes closer than anticipated which will result in a coarse surface).

The reduced surface is generated by virtually rolling the probe along the intersections between the atoms on the surface of the molecule. In general this will create some concave spherical triangles, where three spheres intersect and some torus sweeps at the intersection between two spheres. These created faces must be converted into Bèzier patches.

PDB Id	# atoms	t_0	t_1	t_2	t_3	t_4	# patches
1A3J	166	0.08	0.02	0.03	0.03	0.14	1,848
1A3I	171	0.08	0.02	0.03	0.03	0.14	2,101
100D	489	0.27	0.02	0.09	0.09	0.47	6,346
101D	556	0.33	0.01	0.09	0.08	0.52	6,281
103D	772	0.55	0.02	0.14	0.11	0.81	7,496
1HY8	1,158	0.88	0.03	0.19	0.16	1.27	12,163
1GQG	12,136	4.52	0.11	1.03	0.92	6.54	70,965
1DT5	17,081	8.83	0.23	2.22	1.92	13.23	142,744

Table 2: *Running time in seconds for different molecules on an Athlon 1700+ with 512 MB Ram (t_0: build reduced surface, t_1 build analytical surface, t_2 handle special cases, t_3 build subdivis t_4 total time)*

3.5.1. Generating Bézier Patches

The solvent excluded surface can be built with only two different kinds of patch types: a spherical triangle and a torus sweep. These two patch types can easily be generated by subdivision surfaces, as shown in detail below.

3.5.1.1. Sphere Triangle The spherical triangle is a triangle bounded by three vertices and three circle segments s_0, s_1, s_2 connecting these vertices. Each of these circle segments $s_i = (c, p_0, p_2)$ (Figure 13 a) can be converted into a Bézier curve through the following dependencies ([SR04]):

- p_0 and p_2 have bézier weights $w_0 = w_2 = 1$.
- The control point p_1 lies on the line $\overline{cm}$ with $m = \frac{1}{2}(p_0 + p_2)$, distance $d = \frac{r}{cos\alpha}$ from c and weight $w_1 = cos\alpha$.

From these three Bézier curves a patch generating a spherical triangle can be obtained by connecting the three control points as shown in Figure 13 b).

3.5.1.2. Torus Sweep The torus sweep is a quad bounded by four circle segments s_0, s_1, s_2, s_3 as shown in Figure 13 c). From these four segments only two are needed for the sweep, namely s_0 and s_1. From the generation process we know that:

$$\langle p_3 - p_0, (p_0 - p_1) \times (p_2 - p_1) \rangle = 0$$

Therefore it is possible to rotate and translate the segments so that s_1 lies in the *zx*-plane, s_0 lies in the *xy*-plane and the mid point of s_0 lies in the point of origin. Now it is very easy to generate p_4, p_7 and p_5, p_8 by just scaling s_0 to the desired radius given by p_1 and p_2. p_0, p_1, p_2, p_6, p_7 and p_8 all have weight one. p_3, p_4 and p_5 have a weight according to the formula above. The patch can be obtained by building four quads like in the Figure 13 c).

The generated patches can, for example, be subdivided by the de Casteljau algorithm [dC59, dC63].

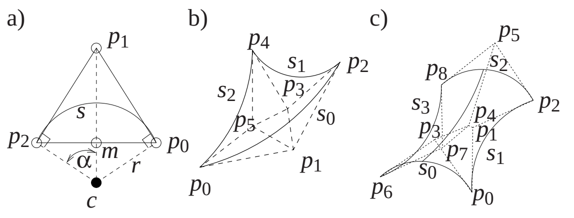

Figure 13: *(a) Calculating a Bézier Curve (b) Generating a triangular Bézier patch (c) Generating a quadrangular Bézier patch*

3.5.2. Results for Surfaces

As shown in Table 2 the complete running time of the surface calculation is more or less linear to the number of atoms. This is optimal since the number of faces in the reduced surface is in $O[n]$ as shown by Sanner in [San92]. The number of patches created for the surface is in $O[n]$, too. The most time-consuming part of the surface calculation is the generation of the reduced surface, followed by the handling of special cases and the generation of the subdivision mesh. But the overall computation time is short enough for even the biggest molecules.

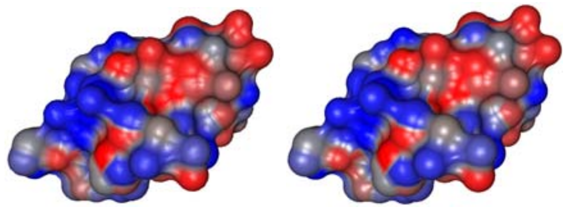

Figure 14: *Generated solvent excluded surface for protein* 1a3i. *On the left with adaptive tessellation and on the right with full resolution.*

4. Conclusion and Future Work

In this paper we presented fast visualization concepts for molecular structures. These concepts are based on multiresolution meshes, subdivision surfaces and modern graphic boards extensions. Therefore, these concepts are not limited to visualization problems in biology but can also be used in other fields. For example, the billboard idea might be used in astronomy, when visualizing thousands of stars or in other particle systems ([BDST04, HE03]). Also, the Combined BReps together with a sophisticated method to generate the base mesh is extremely efficient and can easily be adapted for other purposes.

By using these techniques, we developed the tool *BioBrowser*, which can be used for interactive navigation through the proteins as well as for the production of high quality images for publications or presentations. In the past this was a two step procedure: first navigate to the desired position with one tool, then saving this position and load it to another tool for generating the high quality image.

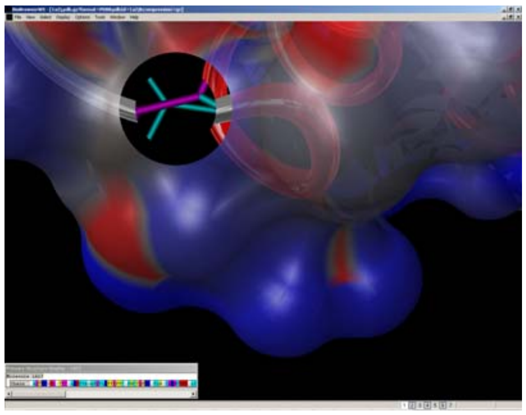

Figure 15: *First impression of a semantic lens*

The next steps will implement new visualization modules. For example, the bond model will be replaced by a new model using billboards and fragment shaders, similar to the space-fill model. Another useful feature will be a semantic lens which will help the user to switch between the different visualization styles and to combine two or more styles in one molecule, e.g., combining a half-transparent surface with the ribbon and stick style. Figure 15 gives an idea of what it will look like. With all the necessary visualization features available at interactive rates, even for extremely complex proteins, we will be able to 'upgrade' the bare visualization tool into a tool which allows a researcher to embed and to retrieve additional bio-information, typically textual, as well as hyperlinks directly into the 3D molecular structures – similar to what current web browsers offer in the textual domain. Speaking of web browsers: of course, the *BioBrowser* is also available as a browser plug-in for visualizing molecules over the Internet.

Acknowledgment

The support of this project by the German Research Foundation (DFG) under grants Fe431/4-3 and Fe431/6-1 is gratefully acknowledged.

We also want to thank Sven Havemann for his CBrep Library 'Mesh-Lib' and the support he gave us, and Lars Reusche, who helped us with the subdivision-surfaces in the surface model.

References

[BDST04] BAJAJ C., DJEU P., SIDDAVANAHALLI V., THANE A.: Texmol: Interactive visual exploration of large flexible multi-component molecular complexes. In *VIS '04: Proceedings of the IEEE Visualization 2004 (VIS'04)* (2004), IEEE Computer Society, pp. 243–250. 7

[BLMP97] BAJAJ C., LEE H., MERKERT R., PASCUCCI V.: NURBS based B-Rep models for macromolecules and their properties. *Proceedings of the 4th Symposium on Solid Modeling and Applications* (May 1997), 217–228. 6

[BPS*99] BAJAJ C. L., PASCUCCI V., SHAMIR A., HOLT R. J., NETRAVALI A. N.: *Multiresolution Molecular Shapes*. Tech. Rep. 99-42, TICAM, 1999.

[Car87] CARSON M.: Ribbon models of macromolecules. *J.Mol.Graphics* (1987), 103–106. 4

[Car91] CARSON M.: Ribbons 2.0. *J.Appl.Cryst.* (1991), 958–961. 4

[Car97] CARSON M.: Ribbons. *Methods in Enzymology* (1997), 493–505. 4

[CB86] CARSON M., BUGG C.: Algorithm for ribbon models of proteins. *J.Mol.Graphics* (1986), 121–122. 4

[CC78] CATMULL E., CLARK J.: Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* (September 1978), 350–355. 4

[Chi] Chimera. http://www.cgl.ucsf.edu/chimera/. 1, 4

[cn3] Cn3D. ftp://ftp.ncbi.nih.gov/cn3d/. 1, 4

[CWWS03] CAN T., WANG Y., WANG Y.-F., SU J.: Fpv: fast protein visualization using java 3D^{TM}. In *Proceedings of the 2003 ACM symposium on Applied computing* (2003), ACM Press, pp. 88–95. 1

[dC59] DE CASTELJAU P.: *Outillages méthodes calcul*. Tech. rep., A. Citroen, Paris, 1959. 7

[dC63] DE CASTELJAU P.: *Courbes et surfaces à poles*. Tech. rep., A. Citroen, Paris, 1963. 7

[Ede95] EDELSBRUNNER H.: The union of balls and its dual shape. *Discr. Comput. Geom. 13* (1995), 415–440.

[EM94] EDELSBRUNNER H., MÜCKE E. P.: Three-dimensional alpha shapes. *ACM Transactions on Graphics 13*, 1 (1994), 43–72. 6

[Far90] FARIN G.: *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, Inc., 1990. 4

[GB78] GREER J., BUSH B.: Macromolecular shape and surface maps by solvent exclusion. In *Proc. Natl. Acad. Sci. USA* (1978), vol. 75, pp. 303–307. 6

[Hav02] HAVEMANN S.: Interactive rendering of catmull/clark surfaces with crease edges. *The Visual Computer* (2002), 286–298. 4

[HE03] HOPF M., ERTL T.: Hierarchical Splatting of Scattered Data. In *Procceedings of IEEE Visualization '03* (2003), IEEE. 7

[HF03a] HAVEMANN S., FELLNER D.: *Generative Mesh Modeling*. Tech. Rep. TUBSCG-2003-01, Institute of Computer Graphics, TU Braunschweig, 2003.

[HF03b] HAVEMANN S., FELLNER D.: *Progressive Combined BReps – Multi-Resolution Meshes for Incremental Real-time Shape Manipulation*. Tech. Rep. TUBSCG-2003-05, Institute of Computer Graphics, TU Braunschweig, 2003. 4

[HOF04] HALM A., OFFEN L., FELLNER D.: Visualization of complex molecular ribbon structures at interactive rates. In *Proceedings of the Information Visualisation, Eighth International Conference on (IV'04)* (2004), IEEE Computer Society, pp. 737–744. 4

[KHM*03] KREYLOS O., HAMANN B., MAX N. L., CRIVELLI S. N., BETHEL E. W.: Interactive protein manipulation. In *Proceedings of the 14th IEEE Visualization Conference 2003* (2003). 1

[LR71] LEE B., RICHARDS F. M.: The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Biol. 55* (1971), 379–400. 5

[pdb] RCBS protein database. http://www.pdb.org. 1, 2

[ras] Rasmol. http://www.umass.edu/microbio/rasmol/. 1, 4

[Ric77] RICHARDS F.: Areas, volumes, packing and protein structure. *Annu. Rev. Biophys. Bioeng. 6* (1977), 151–176. 6

[Ric81] RICHARDSON J.: The anatomy and taxonomy of protein structure. *Adv. Protein Chem.* (1981), 167–339. 2, 4

[San92] SANNER M.: *Modeling and applications of molecular surfaces*. PhD thesis, Université de Haute-Alsace (France), 1992. 6, 7

[SGwHS98] SHADE J., GORTLER S., WEI HE L., SZELISKI R.: Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM Press, pp. 231–242. 4

[SO96] SANNER M. F., OLSON A. J.: Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers 38* (1996), 305–320. 6

[SR04] SÁNCHEZ-REYES J.: Geometric recipes for constructing bézier conics of given centre or focus. *Computer Aided Geometric Design 21* (2004), 111–116. 7