Mapping Volumetric Properties on Molecular Surfaces in Real-Time

Michael Teschner and Christian Henn

Silicon Graphics Inc.
Chemical, Pharmaceutical and Medical Industry Marketing
4125 Riehen, Switzerland

Abstract

A method for the fast generation of solid molecular surfaces is described. Based on a three dimensional grid, a template driven procedure generates the so called "contact surface", also known as Connolly surface. The main advantages of this method are its good performance and that it directly generates continuously tessellated geometry.

Texture mapping is a powerful computer graphics procedure, which can be used to visualize molecular properties on top of solid molecular surfaces. By using texture mapping, traditional color coding can be extended to multidimensional color coding, information filtering and property clipping. Three dimensional textures can be used for interactive color updates on a moving surface, representing the property of a surrounding static scalar field.

1 Introduction

Interactive visualization of molecular properties and real-time manipulation of display-parameters are still among the most important problems in molecular graphics applications. As the amount of data accumulated by experiments and simulations grow at an incredible speed, there is an increasing need to analyze relevant molecular properties by interactive inspection of 3D molecular scenarios.

Years ago, surfaces have been identified as a powerful visualization tool for molecular graphics, both to represent molecular shape [1] and to extract contour information from volumetric property grids [2]. Both types of surfaces have been composed of dots or polygonal meshes, the only suitable way for vector based graphics hardware. This technique works well for the interactive study of small and specific regions, but complex images become difficult to interpret.

With today's raster technology it is possible to represent molecular surfaces in a space-filling mode. Through shading and lighting calculations one can obtain a better three dimensional understanding of the displayed scenarios, as opposed to the chicken wire display. There are many possibilities for the generation of raster based images. However, one has to find a compromise between high quality pictures and performance.

The representation of surfaces by a set of polygons is much faster than any pixel oriented technique. Triangles [3,4] have been found to show a number of advantages compared to other polygons. There are fast algorithms [9] in use to start from 3-D grid with scalar values and generating isosurfaces based on triangles. However, the main challenge remains the fast generation of solid surfaces based only on geometric data, such as atom positions and atomic radii.

3-D grids are used to represent molecular properties in various ways. These grids can also be used to describe molecular geometry, as structure and topology. The advantage of grid based methods to describe molecular geometry is the direct mapping of world coordinates into a 3-D memory (the grid). This results in efficient and fast algorithms, especially for large systems. The average performance decrease is only linear to the number of atoms. Since there is no long-range dependency of geometric data, this is also very well suited to run in parallel mode. A third advantage is that some of the strategies, already developed to visualize 3-D data sets, can be used and modified in order to visualize molecular geometric data.

2 Molecular surfaces and volumetric properties

Many molecular properties are observed as scalar data values sampled over a grid in three dimensions. Examples for this volumetric data include electron densities from
X-ray crystallography, 3-D reconstruction from electron microscopy and electrostatic potentials calculated around molecules. There are two different visualization methods for the display of 3-D maps.

The volume rendering method displays the map directly as a continuum. The values in the map are translated into different color and transparency values. This is often combined with look-up table techniques in order to improve the visual appearance. The result can be compared with a cloud, which can, with the help of the computer, be investigated from different view angles. Implementations of these direct methods are mainly pixel based. The data are sampled while a ray is passing through the grid [5,6]. The main problem with this strategy is its poor performance. It takes several minutes on a modern workstation to render one frame. Recently, hardware-supported texture mapping has been applied for the first time to achieve real-time volume rendering [7,8].

The iso-contouring method is an indirect method, which uses geometry to display discrete regions of the map. The isosurface is defined by points of identical map values. An iso-contour built of triangles can be generated by separating the grid points with function values above the selected contour values from those below. One particular algorithm, the 'marching cube', was originally developed to visualize computer tomography data [9,10,11]. For every cube, formed of 8 neighboring points, one can find a set of connected triangles separating the grid points according to their values. (Figure 1). The number of possible patterns is limited to 256. Due to rotational and inversion symmetries, this number can be reduced further to a small number of basic patterns. These define the basic building blocks for the surface construction. An important fact is that the cubes can be treated independently from each other.

A second class of molecular surfaces is purely based on a geometric description of the molecule. Starting from atomic coordinates and radii, the surface is defined as an envelope to the molecule described as a hard-sphere model. In this article such geometry based surfaces are referred to as "molecular surfaces", as opposed to the "iso-surfaces" described above.

The detailed study of molecular surfaces has become an important tool for the interpretation of properties beyond the molecular structure in terms of atomic coordinates [12,13]. Cavities and channels play an important role in the structure and function of proteins. Investigating the molecular shape of proteins may be useful when designing drugs or more general, ligands interacting with proteins. Color coding molecular potentials onto surfaces provide insight how molecules interact [14].

The basic idea of generating this type of surface is to roll a probe sphere over the hard-sphere model of the molecule. The resulting surface has two different types of regions. Contact regions, where the probe has contact with only one atom and reentrant regions, where the probe has contact with more than one atom (Figure 2).

If the surface is described through the center of the probe sphere then the surface is called solvent accessible surface [15]. Otherwise the surface is called contact surface or Connolly surface. By varying the probe sphere one can model different solvents.

The conventional way of representing the molecular surface is the display of a set of points, describing the surface.
It has been shown, that triangular meshes can be generated from dotted surface representations with no other information than the 3D coordinates of the surface points [16]. However, this postprocessing step can be very time consuming and cannot be applied to complex systems such as zeolites.

There are other attempts to calculate molecular surfaces by using the existing iso-contour algorithms. The 3-D grid represents either simple distance-to-atom information [17], or more sophisticated atomic potentials [10]. In both cases, the reentrant parts of the surface are not handled properly.

3 Fast generation of molecular surface geometry

The proposed method to generate contact surfaces combines the performance and simplicity of grid based approaches with the geometric accuracy of traditional analytic molecular surface construction. The information requirements are identical to any other hard-sphere model: atomic position and radius for each atom.

In principle, the grid stores the distance of each grid point to the nearest atom. Similar to the marching cube an iso-contour is constructed, which represents the center of the probe sphere rolling over the molecule. This iso-surface is only a rough approximation constrained to the grid, which is then projected onto the atoms to reveal the actual contact surface. Pieces of the surface contributing to the reentrant part are replaced by precomputed surface segments.

A 2-D model serves best as a guide to the surface construction procedure. Here, the molecular shape will be represented by a set of circles. The problem now is to obtain an iso-contour line revealing contact between a probe disk and the circles.

The first step defines a grid with an appropriate point to point distance. This grid-spacing is the parameter influencing the quality, i.e. the number of line segments of the final iso-contour. The size of the grid has to extend the bounding box of the molecule with twice the probe radius along all axes. Next, the distance to the nearest atom is calculated for each grid point and both informations, distance and atom number, are kept. This step completes the preprocessing.

For the following steps, four neighboring grid points forming a square are being considered as the key data structure.

Rolling the probe disk around the atoms results in a trajectory of the probe center. All squares where the trajectory of the center passes through will be used for the contour generation, all the other squares are ignored during the procedure (Figure 3).

- Probe sphere

Figure 3: Projected molecular shape with the center trajectory of the probe sphere rolling around the shape. The displayed points are building the squares used to construct the surface.

Looking at the remaining squares, three different patterns of grid point locations with respect to this trajectory can be identified, as some of the grid points will be inside, and others will be outside of the trajectory. This pattern can be used as a template index to define the segment connectivity of the iso-contour passing through the grid point square.

If the trajectory passes twice through the square, i.e. by tangentialing two different atoms, the square will define two independent patterns, requiring special treatment. Patterns with one vertex lying inside can be neglected, because the neighboring squares take care of this vertex. The patterns with two and three vertices inside the trajectory make up a line segment (Figure 4).

Figure 4: Patterns for the description of base-points (left), corresponding connections of the contour grid (right).
Figure 5 shows the resulting contour after drawing the patterns. This contour is a rough approximation of the trajectory curve and is constructed from vectors connecting points of the atom distance grid. During the next step, each vertex is projected onto the nearest circle, using the distance and direction to the atom center.

Starting from the 2-D case, the extension to 3-D is straightforward. Instead of squares, the main computational focus regards eight neighboring grid points forming a cube. The line segments of the 2-D case get extended to triangle strips, coded for by seven distinct inside/outside patterns. And for the shared reentrant part, precomputed triangle sets are added to the projected geometry, to account both for the saddle and concave contact surface areas, for two, respectively three intersecting atoms.

The result is a continuously tessellated surface composed of triangles, directly computed from the geometric data. The triangles are all about the same size and carry information to which atom they belong and if they are part of the reentrant or contact part of the surface as well.

The quality of the surface, that means the degree of tessellation, can be manipulated by the grid spacing.

Void of long-range geometric context, the algorithm can be easily implemented on parallel systems.

4 Texture mapping

Texture mapping is a computer graphics technique, which was developed for visual simulation and animation applications [18]. But in a more abstract and general definition, texture mapping provides a powerful and elegant framework for the display and analysis of technical and scientific data and can be applied to a variety of different application areas, such as CAD, Chemistry and Medicine.

The texture mapping being described in the following paragraphs is based on the implementation of IRIS GL on the RealityEngine architecture of SiliconGraphics [19].

Three basic components are needed for the texture mapping procedure: (1) the texture, which is defined in the texture space, (2) the 3-D geometry, defined on a per vertex basis and (3) a mapping function that links the texture to the vertex description of the 3-D object.

The texture space is a parametric coordinate system which can be 1, 2 or 3 dimensional. Analogous to the pixel (picture element) in screen space, each element in texture space is called texel (texture element). The texture space may be interpreted as a special memory segment, where a variety of information can be stored that can be linked to a 3-D object through a user specified mapping.

The mapping procedure assigns a coordinate in texture space to each vertex of the 3-D object. It is important to note that the dimensionality of the texture space is independent from the dimensionality of the displayed object.

Before being mapped, the texture coordinates are multiplied by a 4x4 transformation matrix, which is by...
default set to identity. By modifying this texture matrix, one can make the texture slide over the surface, rotate, stretch, shrink and project, or any combination thereof.

Values in the texture map can be used as colors directly, or they can modulate the default color of the surface. Latter example combines the lighting calculations with the texture mapping in one rendering pass.

Two main application areas will be described here: (1) how to add information to geometric object descriptions, and (2) how to visualize volumetric properties.

4.1 Color coding, information filtering and property clipping

Color coding of arbitrarily shaped surfaces may serve as a good starting point to understand the flexibility and power of texture mapping.

The traditional way of color coding uses the Gouraud shading model. RGB color triplets are assigned to the vertices of the 3-D geometry based on the computed property. The pixel colors are linearly interpolated in RGB space, leading to inaccurate results (Figure 7, left). With a highly tessellated surface, this problem can be reduced. An alignment of the surface vertices with the expected color code change or multi-pass rendering will remove such artifacts completely. However, large numbers of polygons or algorithmic complexity is the immediate consequence of such an approach.

Texture mapping can easily solve this problem using a color ramp represented as a 1-D texture. In contrast to the above described procedure, the scalar property information is used directly as the texture coordinate being assigned to surface vertices. The interpolation is then performed in texture space (Figure 7, right). The color is evaluated at every pixel through linear interpolation in texture space. High contrast variation in the color code is now possible, even on sparsely tessellated surfaces. The benefits of using this approach is demonstrated in Figure 8, which shows the contact surface of Ethanol color coded against the electrostatic potential (ESP), using traditional RGB color coding (left) and texture mapping (right). With texture mapping, the resulting sharp transitions from one color value to the next significantly improve the accuracy of rendering. In addition, the sharp transitions help to visually understand the object's 3-D shape.

Figure 7: Difference between interpolation in RGB color space (left) and in one-dimensional texture space (right).

Figure 8: Electrostatic potential (ESP) coded on the contact surface of Ethanol, using RGB shading (left) and texture mapping (right).

It is important to note that, although the texture is one-dimensional, it is possible to tackle a 3-D problem. The dimensionality of the texture space and the object space is independent, thus they do not affect each other.

This feature of texture mapping, as well as the difference between texture interpolation and RGB color interpolation is fundamental to understand the examples presented in this article.

The concept of visualizing a property onto a molecular surface using a 1-D texture can be extended to two or even three dimensions. Each dimension can be defined and used independent from the other. This provides the capability to combine six dimensions of information to describe an object: three dimensions for the shape of the surface (x, y, z) and three more as color information from texture mapping. However, care must be taken not to overload the surface with too much information.

This can be illustrated with Gramicidin A as an example. Gramicidin A is a channel-forming polypeptide controlling the transport of ions through membranes. In its biologically active form it acts as a dimer. For the purpose of this paper, the monomeric unit is sufficient. Figure 9 shows atomic structure and contact surface of Gramicidin A as described in Chapter 2.
Two independent molecular properties provide insight into the biophysical behavior of the molecule. The ESP describes how different surface parts of the molecule interact with a positive probe charge. The property is computed using a point-charge model for the atoms and the coulombic equation for the points [20]. The molecular lipophilic potential (MLP) identifies hydrophobic regions of the molecule. The MLP can be estimated using an empirical method to assign a contribution from each atom to the partition coefficient log P and a distance function for each point [21, 22].

It would be very useful to combine the visualization of both properties on one surface, however, it is very difficult to generate a 2-D color ramp providing a clear insight into both properties at the same time.

Figure 10: Color coding of the Gramicidin A contact surface based on the ESP. The MLP is applied as a filter to blank out all hydrophobic areas.

A solution to this problem is to use a 2-D texture map. In the first texture space dimension the ESP is represented as a color code. The second dimension is used for the MLP, translated into a delta type function that makes the MLP act as a filter for the ESP. As a result, the ESP is color coded only in hydrophilic regions of the surface. At the hydrophobic regions, the color code is blanked out to white (Figure 10). By applying different scaling factors in texture space it is possible to interactively change the filter.

Figure 11: Same as Figure 10, using transparency instead of white color in the texture map, clipping away the hydrophobic parts of the surface.
An extension of the filtering is the surface clipping, that can be achieved by adding to the white portions of the texture complete transparency (Figure 11). With this powerful mechanism, it is possible to use any kind of computed property as a clipping criterion for an arbitrarily shaped molecular surface. Furthermore, it is important to note, that the clipping is not done by changing the data structure, but by applying a texture containing transparency values. Similar to the previously described filtering example, the clipping can interactively be changed through scaling and translation in the texture space.

4.2 Volumetric properties using 3-D texture mapping

In the previous chapter, texture mapping adds information to molecular surfaces, where the texture space is used as a lookup table. This lookup table contains color information or transparency values. The properties being visualized are calculated for each vertex of the surface. We may call this an indirect method to visualize molecular properties. There are up to three independent information channels according to the three dimensions in texture space.

A more natural way to use a 3-D texture space is to represent the three dimensional data directly. Many molecular properties are available as scalar values sampled over a grid in three dimensions, or in other terms, as volumetric data. This data can be transferred to texture memory as a 3-D texture. Strightforward vertex based geometry can be used for direct volumetric representation. The final coloring of the geometry reflects the result of its intersection with the texture. To find the color for each pixel, linear texture space interpolation is done by the hardware.

The simplest application of this technique is that of a slice plane, defined either in object or world coordinates. The slice plane samples the texture to reflect the contents of the volume as if it were exposed by cutting the object with a knife (Figure 12). The orientation of such a slice plane may be modified interactively.

This type of volume visualization is greatly enhanced by using texture lookup tables. In this way, it becomes possible to interactively modify the transfer function which translates the texture values into color and transparency information.

Systematic extension of the slice plane approach leads to the complete visualization of the entire volume. A stack of planes, oriented parallel to the computer screen, samples the entire 3-D texture at regularly spaced and sufficiently small intervals. The planes are drawn in back to front rendering order [7].

The concept of the slice plane may also be extended to arbitrarily shaped objects. The idea is to probe a volumetric property and to display it wherever the surface of the probing object intersects the volume. The probing object can be of any shape, e.g. a sphere that collects information about the volumetric property at a certain distance from a specified point, even a complete molecular surface may serve as the probing device.

The independence of the object's transformation from that of the 3D volume, treated as a 3-D texture, offers complete freedom in orienting the surface with respect to the volume, providing an opportunity to look at a molecular surface and have the mapped property updated based on the current orientation of the object with respect to the volume displayed.

5 Conclusion

Texture mapping as an advanced computer graphics technology with a rapidly increasing general availability may be combined with an efficient surface generation algorithm to provide a powerful innovative framework for the next generation molecular visualization applications. Due to the hardware support of texture mapping, implementation of new modeling functionality becomes straightforward. As the presented techniques complement existing modeling functionality to visualize non structural properties of molecules in an elegant and efficient manner, a better understanding of complex 3-D molecular scenarios may be achieved within the near future.

6 References


