Visualisation in the SPROUT molecular design program

A. Peter Johnson and Zsolt Zsoldos*
ICAMS, School of Chemistry, Univ. of Leeds, LS2 9JT, U.K.
e-mail: johnson@mi.leeds.ac.uk, zsolt@mi.leeds.ac.uk

Abstract

SPROUT is an interactive computer system for structure based molecular design. The system consists of several modules that address the different subproblems of structure based drug design. This paper describes the visualisation techniques applied in the program: the display of the novel (geometric region) representation of the interaction sites and the molecular surface display based on a 3D grid representation of the cavity.

The hydrogen bonding regions are represented by set operations (subtraction and intersection) of simple spherical and conical 3D objects (with given radii and opening angle) Some complex hydrogen bonding regions are represented by intersections of six or more basic objects. A method for calculating a triangular mesh representation (with normal vectors) of the analytical surfaces of the objects, that have sharp edges and corners because of the intersectioons, is presented in the paper. The geometric parameters of the interaction regions can be changed interactively in which case the surface display is updated real-time.

The volume of space that is available for ligand generation (the cavity of the receptor site) is represented on a 3D grid within SPROUT. The surface of the available space is visualised using an algorithm presented in the paper, that generates a polygonial mesh of the grid points. The grid is also used to cut out stericaly forbidden parts of the interaction site regions. The surface of the reduced object is also visualised using further sphere subtractions.

The presented algorithms are fast, aplicable in interactive visualisation programs. Result images of the rendering of the surfaces, calculated by the algorithms, are demonstrated on examples taken from applications of SPROUT to practical ligand design problems.

1 Introduction

An automatic, interactive computer system, called SPROUT, for *de novo* structure based molecular design is currently under development in the University of Leeds. The system consists of several modules addressing different subproblems of structure based drug design: detection of protein clefts, identification of potential interaction sites, primary molecular structure generation, conversion of primary structures into molecules and analysis of the solutions.

*Presenting author

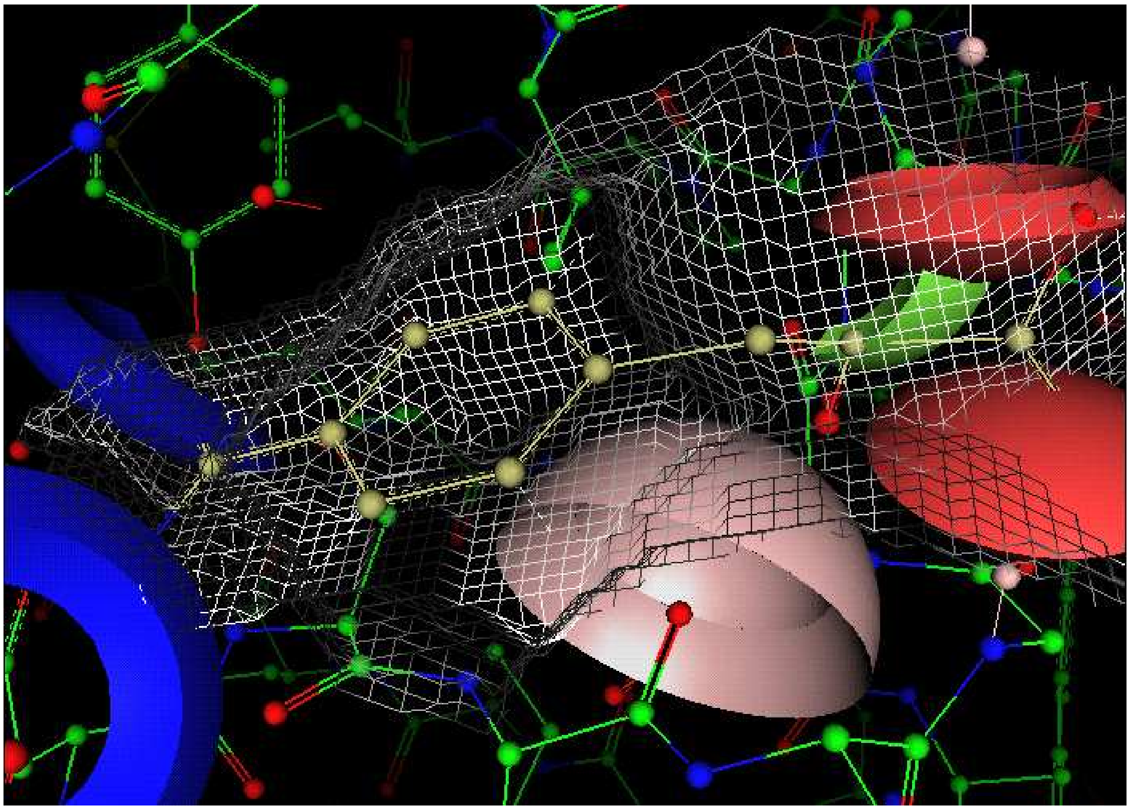

Figure 1: Graphical output example

The interaction site detection uses a rule based approach where the rules are derived from experimental data that was collected from the literature. Typical donor and acceptor atoms are located in the protein and geometric hydrogen bonding regions (according to given distance and angle tolerances) are computed for them.

A 3D graphical visualisation tool has been implemented to display the detected interaction site regions, and the generated chemical structures together with the protein receptor (see Figure 1) and boundary surface. The protein structure and the generated skeletons are represented by atom colored bond-skeletons with small spheres at the atom positions (ball and stick model). The interaction site regions and the boundary surface are represented in three different modes and the user can select interactively which one to be displayed.

The simpliest surfaces representation is a point sampling. A more sophisticated representation is the 'grid wire display' (where the points are connected by lines) and this requires the triangulation of the surface points. A lit solid surface drawing can provide a realistic 3D-like image, generating this requires the definition of surface normals. This paper describes the calculation of the surface points, their triangulation and normal vector calculation.

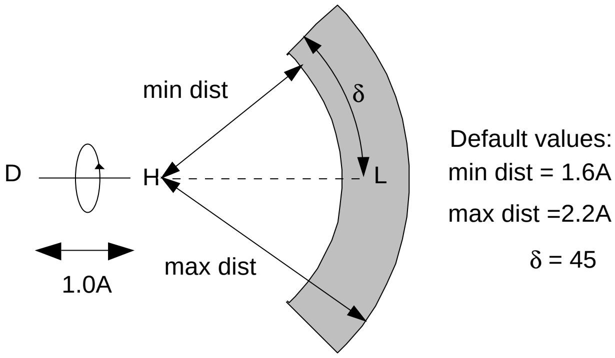

Figure 2: Tolerances for H-bond acceptor

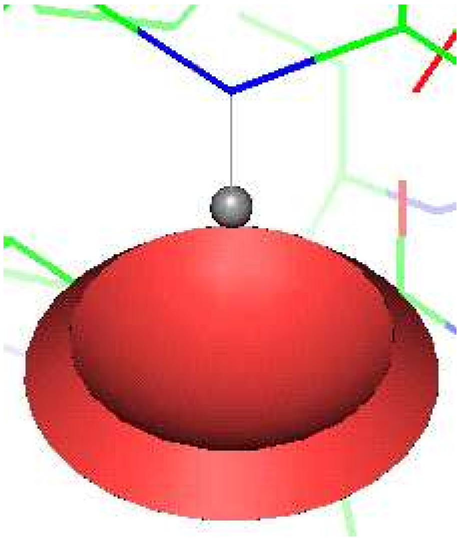

Figure 3: Example for H-bond acceptor region

2 Representation of single target regions

In SPROUT an acceptor target site is defined using the positions of the donor atom and the donated hydrogen of the receptor. An optimal hydrogen bond is formed if the acceptor atom is placed on the straight line defined by the donor atom and the hydrogen atom at about 1.85Å from the hydrogen with a lone electron pair pointing towards the hydrogen. But in reality this ideal geometry cannot be expected in every case, because other atoms may obscure that position or a compromise might be possible by forming another hydrogen bond at the same time (both of them weaker but together stronger than a single ideal bond).

Tolerances are applied to both the distance and to the direction of the ideal bond. The distance tolerance is expressed as an interval, i.e. the minimal and maximal distances between the hydrogen and the acceptor atom are defined. The direction tolerance is defined as an angle difference from the ideal straight direction. The hydrogen bond angle is defined between the hydrogen - donor and hydrogen - acceptor vectors (for the ideal case this angle is $180°$), and so a tolerance of $45°$ means that hydrogen bond angles above $180° - 45° = 135°$ are acceptable. Figure 2 shows a two-dimensional cut of the three dimensional region that the acceptor atom is allowed to occupy according to the distance tolerance values *min dist*, *max dist* and the angle tolerance δ.

The three dimensional region (see Figure 3) is generated by rotating the 2D cut about the donor - hydrogen axis. The surface of the region can be described by the following set operations: take the intersection of a sphere having radius equal to *max dist* with a cone centered on the same point and having δ opening angle. Then subtract a co-centered sphere having radius equal to *min dist* from the object. The surface of the final region can be described as it is built

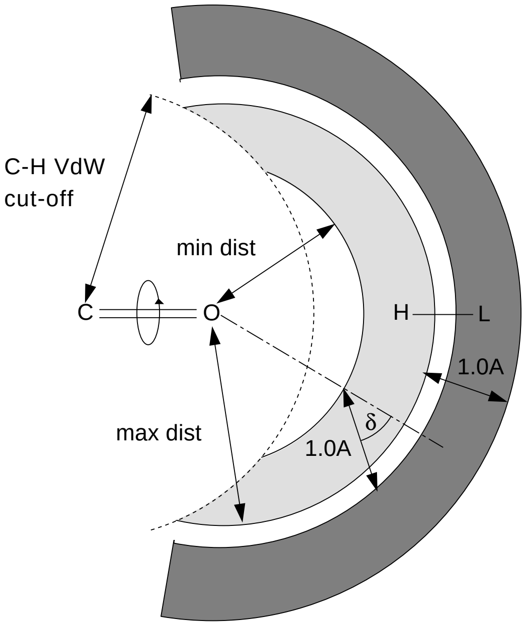

Figure 4: Tolerances for H-bond donor region

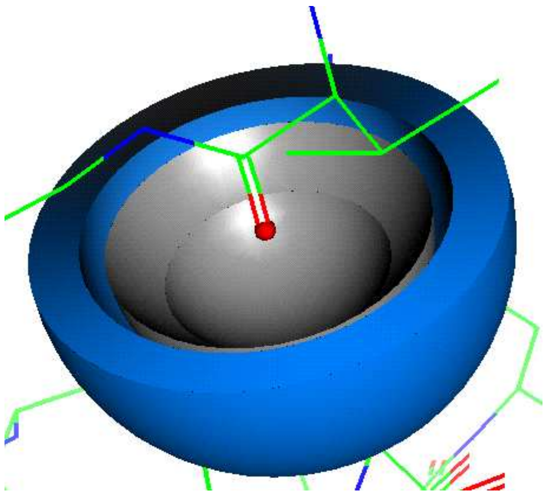

Figure 5: Example for H-bond donor region

up from three pieces: two solid angles (sphere-slice) with the same centre and angle but different radii (the minimum and maximum distance) and a slice of a cone connecting the edges of the solid angles.

If an acceptor ligand atom is placed in any position within the region then a hydrogen bond can be formed between the donor of the receptor and the acceptor of the ligand.

Defining a donor target site is more complicated as it is dependent on the position of the hydrogen. Initially the region where a ligand hydrogen should be placed in order to form a hydrogen bond to an acceptor atom of the receptor is defined. The minimum and maximum distance between the acceptor and the hydrogen defines a sphere layer around the acceptor atom. The possible position of the hydrogen is also limited by any steric constraints of the receptor site. The most obvious steric constraint arises from the adjacent atom(s) of the acceptor, i.e. from the atom(s) that is(are) connected to the acceptor atom. Figure 4 shows an example with a carbonyl group as a hydrogen bond acceptor. The hydrogen must be further from the carbon than the sum of the van der Waals radii of the hydrogen and the carbon. In geometric terms this restriction is described by a sphere centred on the carbon atom with radius that is the sum of the van der Waals radii of the carbon and hydrogen atoms. All the points within this sphere are prohibited positions for the hydrogen, and so this sphere is excluded (subtracted) from the region (which was originated from another sphere).

It is possible to define a donor atom region for every hydrogen position, i.e. for every point within the hydrogen region. This would be a solid angle surface shape with radius 1.0Å and angle δ for each position. The union of these solid angle surfaces for all the hydrogen positions defines the region indicated in Figure 5. The maximum distance between the acceptor and the donor atom is the maximum hydrogen–acceptor (H–O in Figure 5) distance plus 1.0Å (donor–hydrogen bond). The minimum donor–acceptor distance is the minimum hydrogen–acceptor distance plus 1.0Å multiplied by $\cos\delta$. The 2D cross section of the shape of the target site is shown in Figure 4. The 3D shape (see Figure 5) is defined by rotating the shaded areas about the C=O bond.

There are some standard ways of defining spherical surfaces that can be readily applied to molecular surfaces. These methods can be modified easily to define point sampling and triangulation for the surfaces of the single hydrogen bonding regions, because they are built up from sphere and cone parts (the latter have similar geometry to spheres). Some minor problems arise at the normal vector definition for the sharp edges between the parts that originate from different spheres or cones.

2.1 Surface points of single H-site regions

There are different methods for sampling points on a sphere surface and triangulating them. For example, regular polyhedrons, like the icosahedron or the dodecahedron can be used as a basis for an iterative method. Another approach is called the bilinear tessellation, which is based on polar coordinates and samples the points on circles on the sphere. The latter method is selected because it is very simple to use for a solid angle instead of the whole sphere.

The desired sampling density is selected as the number of points per square angstrom. The points are generated on circles that are achieved by cutting the sphere with planes that are parallel to the X and Y axes, i.e. circles on certain latitude values. The points on a particular circle have the same latitude but different longitudes. The latitude angles of the planes are selected by a linear function, i.e. the difference between consecutive latitude angles is constant and this constant is calculated from the density:

$$n_p = \pi R\sqrt{D} \qquad (1) \qquad\qquad N_z = 2\pi R \sin\phi\sqrt{D} \qquad (3)$$

$$\Delta\phi = \frac{\pi}{n_p} \qquad (2) \qquad\qquad \Delta\Theta = \frac{2\pi}{N_z} \qquad (4)$$

In equation (1) R is the radius of the sphere, D is the density, n_p is the number of planes that is necessary for the whole sphere. Equation (2) defines $\Delta\phi$ that is the difference in latitude between the planes. The longitude angle difference is constant between consecutive points on the same circle, and is calculated so to provide the same density. N_z, the number points on the z plane, is calculated using Equation (3). $\Delta\Theta$, the longitude angle difference, is calculated using Equation (4).

The surface sampling for the solid angle δ finishes at the k-th plane, where k fulfils the inequalities (5).

$$(k-1)\Delta\phi < \delta \quad and \quad k\Delta\phi \geq \delta \qquad (5)$$

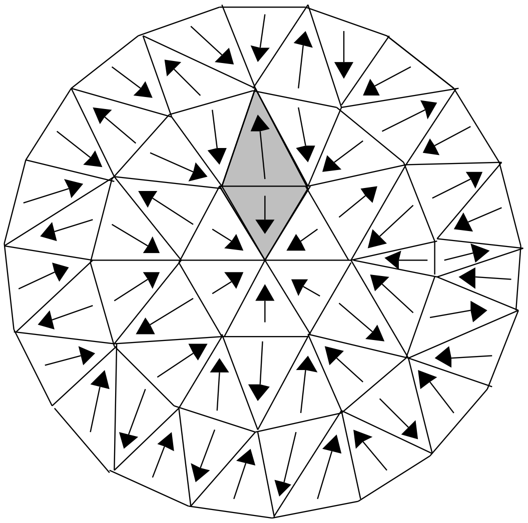

Figure 6: The triangulation method

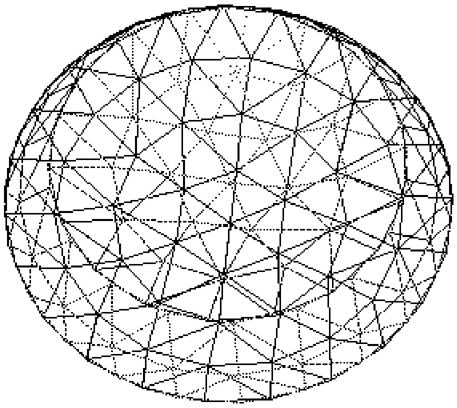

Figure 7: Grid wire display of an acceptor interaction site

The last plane is calculated using latitude angle $\varphi = \delta$ (regardless of the value of $k\Delta\phi$) to get the points of the last circle on the edge of the solid angle.

The cone side surface sampling is done in the same manner using circles on the planar cuts. Similar equations to (1-5) are used, but the radius is changed linearly instead of the latitude angle. The latitude angle is constant δ for each circle (δ is the tolerance value of the interaction site). The constant difference between consecutive radii is $\Delta r = \frac{1}{\sqrt{D}}$.

The first radius is the minimal hydrogen-acceptor distance, the last radius is the maximal hydrogen-acceptor distance, even if the difference is not a multiple of Δr. This ensures that the edge points are generated on both sides of the cone slice.

2.2 Triangulation of the surface points

The sampling points define the shape of the surface, but it is also necessary to define a triangulation of the points to draw a solid surface using a polygonal mesh. An obvious way would be to draw a triangle for every triplet whose vertices are within a threshold defined by the density. This is very inefficient since many areas would be drawn several times.

A real triangulation can be defined using the fact that the point sampling is made on concentric circles with equidistant points. Two triangles can be defined for each segment on a circle. The first one points towards the inner (shorter) neighbouring circle, the other points towards the outer adjacent circle, except for the segments of the innermost and outermost circles which have only one triangle. The third vertex for each triangle is selected from the target circle as the closest point to the centre of the base segment. Figure 6 demonstrates the triangulation for a small example highlighting two triangles that belong to one segment. The arrows show the selection of the third vertices by pointing from the centre of the base segment to the closest point on the target circle. Figure 7 shows the complete triangulation of an acceptor target site that is displayed in grid wire mode.

The time complexity of this triangulation algorithm is a linear function of the number of points. There are as many segments as points, and the triangles are generated by going through the segments once only. It is not necessary to search through the target circle to find the closest third point on the target circle, because the third vertices must be in monotone order for consecutive segments. Thus the search starts on the vertex that the search finished on for the previous segment. The search is stopped when the distance is longer than the previous, because the next distance would be even longer. In this way the distance is calculated at most twice to each point on the target circle. Each circle becomes a target circle twice: once as an inner circle and once as an outer circle. Altogether the distance calculation is done a maximum of four times the number of points.

The triangulation requires linear storage space, because 2 triangles are generated for each segment. It is enough to store 3 indices for each vertex: the end of the segment that starts at the particular point, the index of the inner circle vertex that forms the inner triangle with the segment and finally the index of the outer circle vertex that is used in the outer triangle.

The points on the edges of the surface are duplicated because they are generated on both sides. But it is necessary to duplicate them as they form triangles with very different normal vectors. Having two copies, each of them can have the appropriate normal vector assigned (according to its side) in order to provide the necessary information for the lighting model of the 3D drawing.

3 Representation of intersection regions

The ability to form multicentred and/or bifurcated hydrogen bonds is very important because it has been observed that 25-40% of the hydrogen bonds in biological structures are not simple twocentred bonds but more complex hydrogen bonds [2, 1]. The complex hydrogen bonding cases can be identified by the intersections of the single regions (with broad tolerances, e.g. the default values 1.6Å-2.2Å and 45°). Figure 8 shows and example where a donor and an acceptor H-site intersect. If a hydroxy group resides within this region, it can form two hydrogen bonds acting as a donor and as an acceptor at the same time.

The intersection region of two (or more) simple H-site regions is used for representing compound hydrogen bonding regions, i.e. multicentred and/or bifurcated hydrogen bonding regions. It is easy to test whether a given point is within an intersection of H-bond regions by testing the point against both of them. The indirect question is more difficult to answer, i.e. which points form the intersection region, and hence what is the shape of the intersection. These intersections can have rather complicated shapes, and they all have sharp edges and corners. The points on the edges and corners are not inherited automatically from the sampling of the simple (intersected) parent surfaces, and so they must be calculated. The graphical display requires the triangulation of these surface

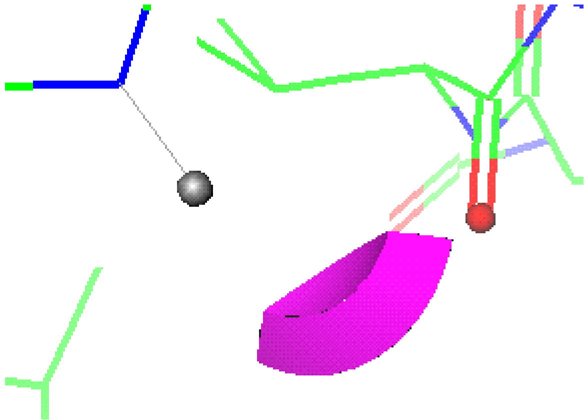

Figure 8: Example for a dual (acceptor and donor) H-bond site

points as well.

3.1 Points on the intersection surface

Finding the intersection points is not only a graphical visualisation question, but in fact, this is the best way to decide which pairs of simple regions intersect; there is no intersection if no points are accepted by the other region. Before the points are checked, a simple distance test is performed for the spherical centres as a primary filter to avoid unnecessary calculations. If the enclosing spheres do intersect, then the intersection points are calculated as follows.

Some points of the intersection surface are selected from the sampling of the parent surfaces. All the surface sampling points of the first region (referred to as the **source region**) are tested to see if they fall within the second region (named the **goal region**). A point falls within the goal region if the following criteria stands. The distance of the point from the centre of the goal region is in between the minimum and maximum allowed distance of the goal region and the angle is within the limit for cone shaped sites (or the point is outside the excluded sphere for sphere-cut shaped sites). The accepted points form one half of the intersection surface. The test is done the other way around as well (swapping the role of the source and goal region), to find the other half of the intersection surface.

As a result the points on the faces of the intersection surface are found, but not the points on the edges and corners, except for some extremely rare and fortunate cases when a surface sampling point of one region is exactly on the surface of the other region. Inheriting the triangulation of the points gives some surface bits on the faces as well. However, there are jagged gaps along the edges of the surface (see Figure 9 for example).

3.2 Points on the edges

In order to generate extra points on the edges, some points that are outside the other region, but have adjacent points inside are searched for. The adjacency

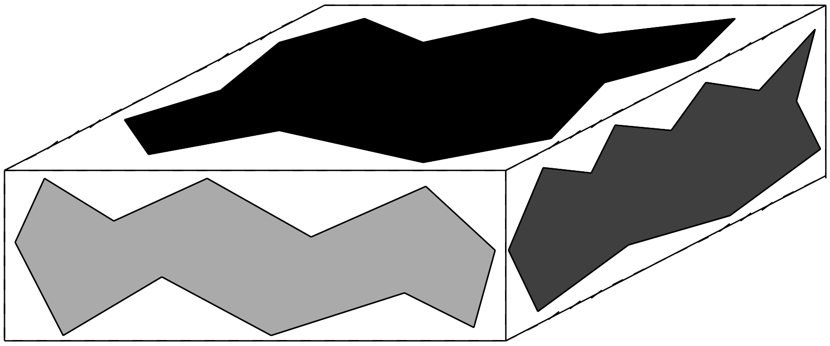

Figure 9: Gaps between the faces of the intersection

is defined according to the triangulation. A **wrong point** is a point outside the goal region that has an adjacent point inside (a **good point**). A triangle edge is known for each of these wrong points that connects it to a good point. The triangle edge lies within the source surface and pokes into (intersects) the goal surface. It is necessary to find the point that is at the intersection of the goal surface and the line connecting the wrong and good points. This type of intersection point is called a **border point**. The border points are calculated by the following procedures and are included in the intersection surface. They inherit the triangulation connections from the wrong point.

There are 3 types of wrong points depending on which tolerance is violated: the distance criterion, the angle criterion or both of them. In the first two cases a border point is generated on the edge of the intersection surface, in the later case a **corner point** can be generated. A corner occurs in a position where a face of one surface intersects an edge of the other surface.

The distance criterion violation means that the wrong point is further than the maximum allowed distance or closer than the minimum allowed distance to the centre of the sphere. Let d_w denote the distance of the wrong point from the centre, d_g is the distance of the good point and r is the limit value (either the maximum or the minimum allowed distance). The intersection point can be computed using a linear interpolation between the wrong and the good points. Equation (6) shows how to calculate the coordinates of the border point B from the coordinates of the good point G and the wrong point W.

$$B_i = G_i \frac{d_w - r}{d_w - d_g} + W_i \frac{r - d_g}{d_w - d_g} \ , \quad \forall i \in \{x, y, z\} \tag{6}$$

The angle criterion violation means that the wrong point has a smaller angle than the limit, i.e. it is beside the cone side slice (Figure 11). A similar linear interpolation to (6) is done but using the angles α_g, α_w and δ instead of the distances d_g, d_w and r respectively:

$$B_i = G_i \frac{\alpha_w - r}{\alpha_w - \alpha_g} + W_i \frac{r - \alpha_g}{\alpha_w - \alpha_g} \ , \quad \forall i \in \{x, y, z\} \tag{7}$$

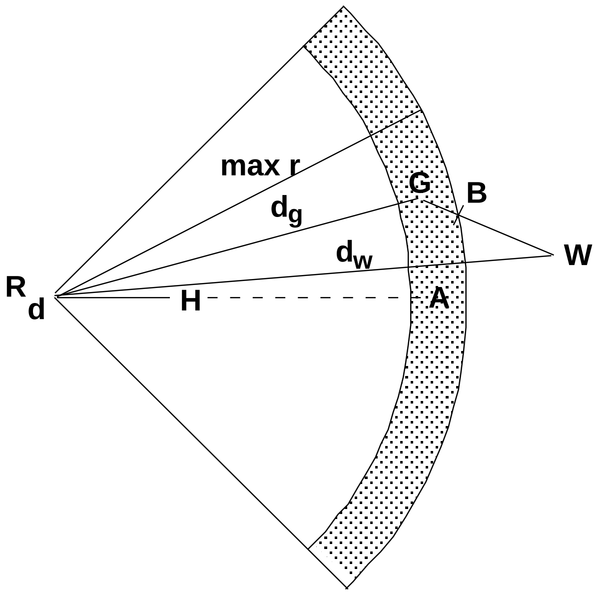

Figure 10: Border point by distance criterion

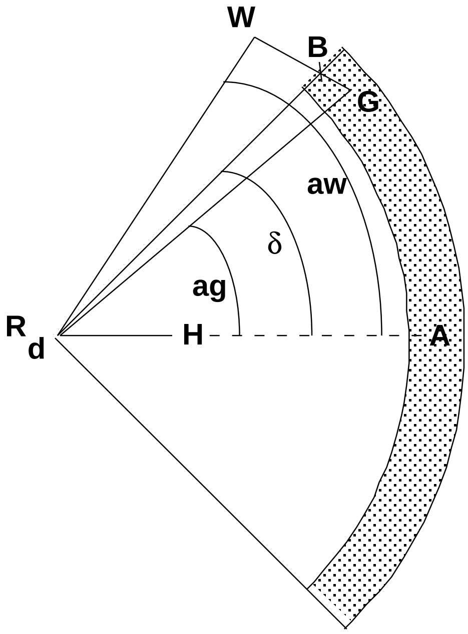

Figure 11: Border point by angle criterion

3.3 Corner points

The situation is more complicated for the double violating corner points because it is not guaranteed that the triangle edge coincides with the corner: $B \neq C$ in Figure 12.

The corner point calculation consists of three steps. The first step is the same linear interpolation as (7) to calculate a border point (B) on the cone side. But this point is still not accepted because of the distance violation. In the second step point C is calculated using equation (8). Point C is accepted by the goal region because it is on the edge of the goal region, but it is probably not on the source surface anymore because it is no longer on the line between the good point and the wrong point. There must be a border point on the same edge of the goal region (as the triangulation follows the edge) that is calculated in the reverse phase, and therefore that point is on the surface of the current source region, but is still on the edge as well, hence it is on the corner. Thus the third step is a search for the closest border point to point C among the border points of the source surface. The coordinates of that corner point are used instead of the coordinates of point C.

$$C_i = R_i + \frac{r}{d_b}\overrightarrow{RB}_i \ , \quad \forall i \in \{x, y, z\} \tag{8}$$

As the edge points of the simple surfaces are duplicated to form triangles of different faces with different normals, the intersection border points are duplicated as well. This happens automatically and the border point calculation is performed for both partial surfaces.

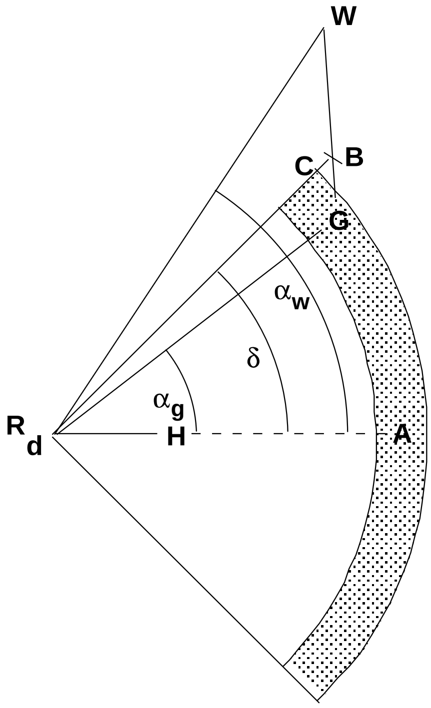

Figure 12: Corner point by two criteria

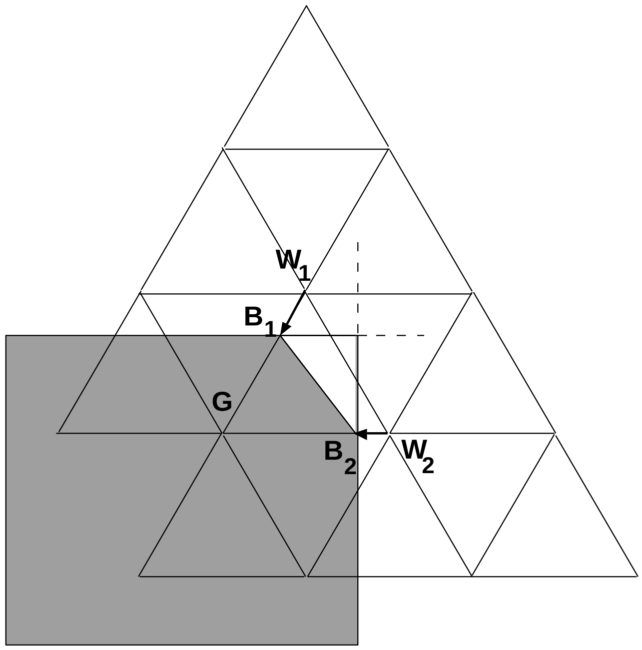

Figure 13: Corner without double violation

The corner points should be generated three times for the same reason, because they belong to three different faces of the surface. Two copies of the corner points are generated with certainty, the ones that originate from the two copies of the edge of one parent surface that intersect with the face of the other parent surface. They are generated as border points from single violations. The third copy might be generated from a double violation on the face surface, but there are cases without double violation at the corner (see Figure 13).

In these cases there are two border points B_1 and B_2 that are connected by the inherited triangulation, but none of their parent wrong points (W_1 and W_2) cause double violation. These cases must be found in order to correct the missing corners. They can be spotted from the connection $B_1 - B_2$ as an edge in the triangulation, which connects border points of different violation types, while connected border points on the same edge have the same violation type (both angle violation or both distance violation). The corner point is calculated using the last two steps of the double violation corner point computation case for the angle violation border point, i.e. a point C is calculated using equation (8), then the closest border point is searched for in the opposite surface half.

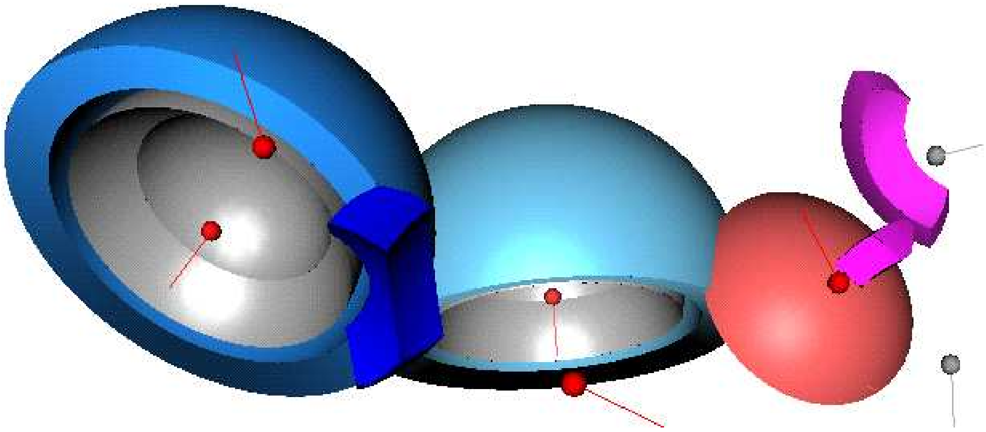

Figure 14: Example for lit solid surface display

4 Surface normal calculation

Realistic 3D graphics can be obtained using lit solid surface drawings (see Figure 14). The Silicon Graphics GL/OpenGL lighting model considers the lighting to be divided into four independent components: emitted, ambient, diffuse and specular. The emitted light originates from the objects (it is not used for molecular surfaces), the ambient component gives a directionless backlighting. The diffuse light comes from one direction, so it is brighter if it comes squarely down on a surface than if it barely glances off the surface. Once it hits a surface however, it is scattered equally in all directions, so it appears equally bright, no matter where the eye is located. Finally, specular light comes from a particular direction, and it tends to bounce off the surface in a preferred direction. One can think of specularity as shininess. The diffuse and specular components require surface normal vectors assigned for each sampling point. The surface normal vectors have unit length and point outwards from the object perpendicular to the local surface part.

The interaction site regions are built from the intersection of spheric and conical objects, consequently the surface normal vectors can be calculated according to the regular object surfaces. The normal vector $\underline{n}_s$ to a point P on the surface of a sphere is calculated using the centre of the sphere C and the radius r using the Equation (9), see also Figure 15. The same calculation is suitable for the normal vectors on the inward face of the sphere-layer part of the interaction sites, the result is multiplied by -1 in this case to obtain the appropriate direction.

$$\underline{n}_s = \frac{\overrightarrow{CP}}{r} \tag{9}$$

The surface normal $\underline{n}_c$ of a point S on a conical object part is calculated using an auxiliary point A (see Figure 16, Equation 11). The coordinates of A are calculated form the coordinates of the tip of the cone R (the receptor atom

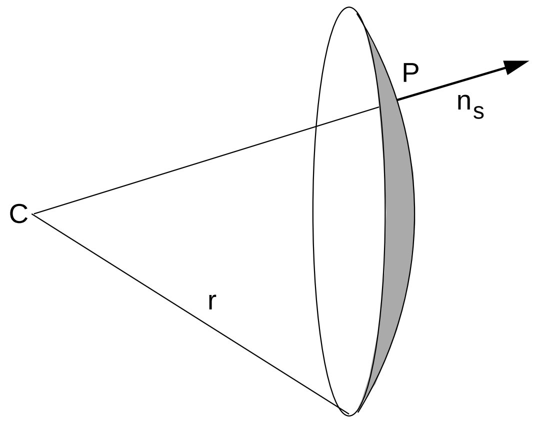

Figure 15: Surface normal of spheric objects

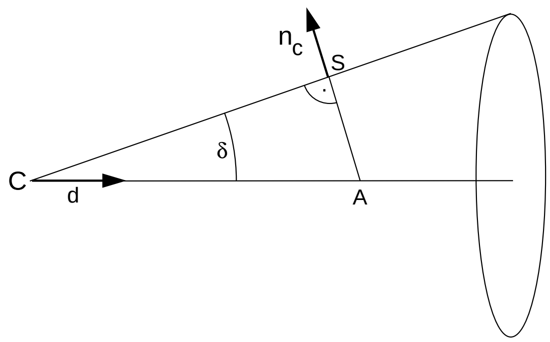

Figure 16: Surface normal of conical objects

position) and the axis direction of the cone d (the donor $\rightarrow$ hydrogen vector) using Equation (10).

$$A_i = R_i + \frac{d_i}{|\underline{d}|}\frac{|\overrightarrow{RS}|}{\cos\delta}, \quad \forall i \in \{x, y, z\} \qquad (10)$$

$$\underline{n}_c = \frac{\overrightarrow{AS}}{|\overrightarrow{AS}|} \qquad (11)$$

5 Boundary grid lines and surface calculation

There is a 3D grid that stores the information on which positions are inside and outside the boundary (cavity), and it also stores the quasi cubic distance from the cavity surface. The cavity surface is represented by grid points for which this distance has zero value. The grid points with cell value zero can be used for dispay purpose as a point sampling of the boundary surface (this is the simpliest surface visualisation). For a grid wire representation, some connections between the points are required. For a solid surface drawing a polygonal mesh and normal vectors are also required.

The connection table calculation for the grid wire display is performed in three phases as follows. (Note, that this algorithm is general, it could be applied to any kind of grid surface as it does not use any information about the spheric nature of the molecular surface.)

The closest possible connection exists between grid points that have two coordinates in common and their third coordinate differs by the grid cell size only. These connections are searched for in the first phase. They result in some canonical (parallel to an axis) lines and plane parts, but the surface is not completely connected.

The second type of connection is between grid points having one coordinate in common and with the other two differing by the grid cell size only. Not all of these connections are necessary because some of them are redundant to the existing first type connections. This happens in cases where the second type

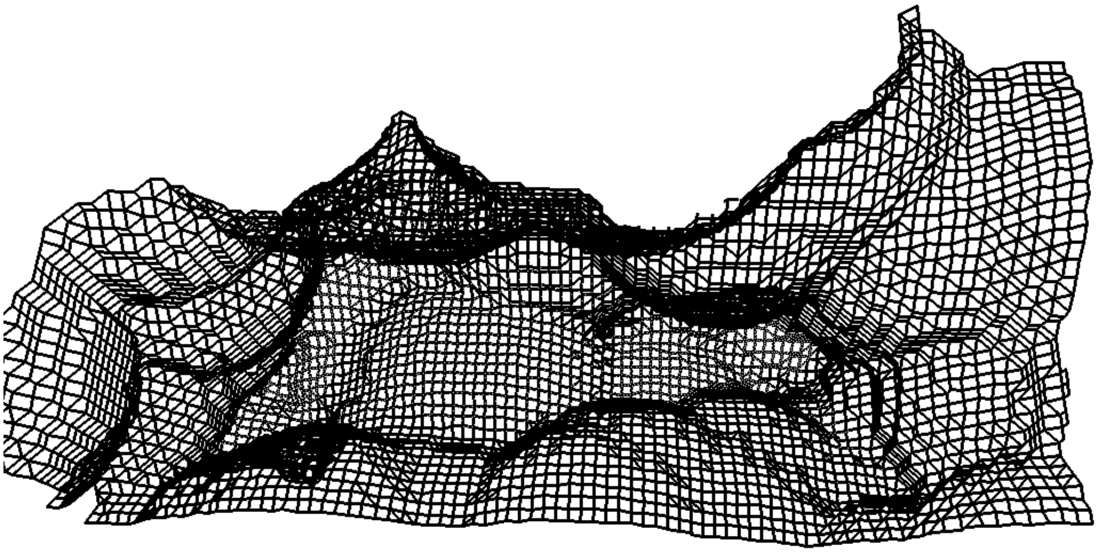

Figure 17: Boundary by grid wire display

edge connects points that are already connected by two first type edges via a third grid point.

A second type connection can be expressed as the vectorial sum of two first type connections. The second type connection is necessary, if and only if, none of its first type connection components are present.

After the necessary second type connections are registered usually the grid wire is complete, except for some extreme concave points that have neither first or second type neighbours. A third phase search is performed to find these cases. The neighbours with no common coordinate but where all of them differ by one grid cell size are selected and they are connected if there is no first or second type connection with the same vector component.

The connections are registered in both directions, i.e. each vertex has the list of its neighbours (a grid point can have a maximum of 6 neighbours). The registered connections can be displayed as line segment to get the grid wire display (see Figure 17). The grid wire consists of triangles, rectangles and pentagons that must be drawn as polygons for solid surface representation. To avoid redrawing the same polygon from each vertex the following rules are applied.

- A triangle is drawn from vertex A to adjacent vertices B and C if, and only if, $A < B < C$ and there is a connection between B and C.
- A rectangle is drawn from vertex A using adjacent vertices B and C if, and only if, $A < B < C$, there is no connection between B and C and there is a common neighbour $D \neq A$ of B and C.
- A pentagon is drawn from vertex A using adjacent vertices B and C if, and only if, $A < B < C$, there is no connection between B and C, there

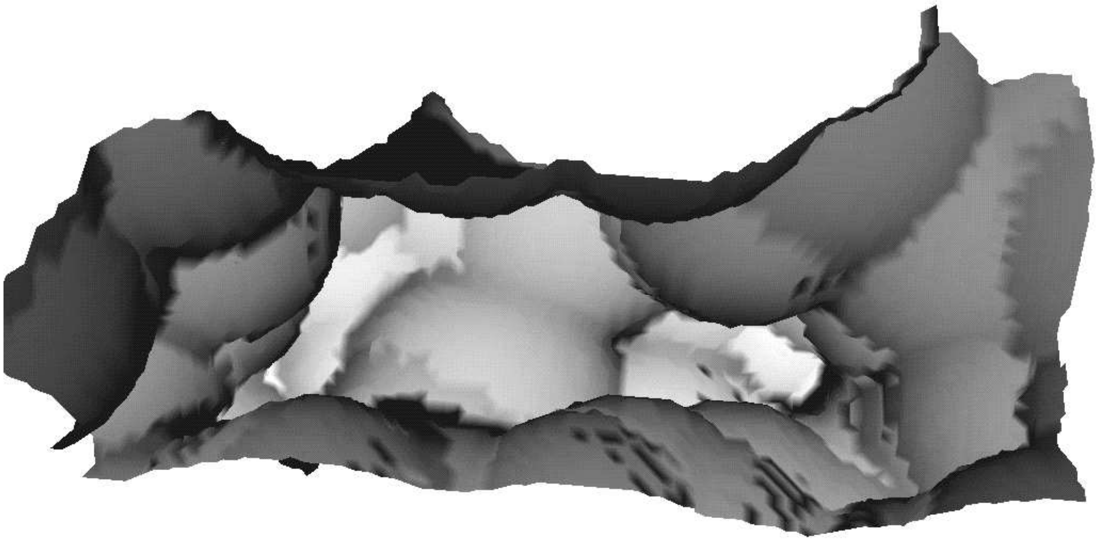

Figure 18: Boundary by solid surface display

is no common neighbour of B and C except A, there is a neighbour D of B which is not connected to A and there is a neighbour E of C which is connected to D but not to A.

The normal vector for each surface grid point is calculated in the direction defined by the vector from the appropriate atom centre to the grid point and normalized to unit length. The appropriate atom centre is the one that is as far from the grid point as its sphere radius (the van der Waals radius plus the probe radius). It cannot be guaranteed that the atom centre is exactly at the same distance as the sphere radius, but the difference must be smaller than the grid cell size.

The grid based cavity surface is a discrete approximation of the spherical solvent accessible surface. Displaying this surface has the advantage that it is exactly the same boundary as the one used by the skeleton generation program (see Figure 18).

References

[1] G.A. Jeffrey and W. Saenger. *Hydrogen Bonding in Biological Structures.* Springer Verlag, Heidelberg, 1991.

[2] R. Taylor, O. Kennard, and W. Versichel. Geometry of the N-H$\cdots$O=C hydrogen bond. 2. three-center ("bifurcated") and four-center ("trifurcated") bonds. *J. Am. Chem. Soc.*, 106:244–248, 1984.