HEWLETT PACKARD

On Generating Topologically Correct Isosurfaces from Uniform Samples

B. K. Natarajan
Software and Systems Laboratory
HPL-91-76
June, 1991

data visualization, three dimensions, isosurfaces

A function $f(x,y,z)$ of three variables may be visualized by examining its isosurfaces $f(x,y,z) = t$ for various values of t. To display these isosurfaces on a graphics device it is desirable to approximate them with piecewise polygonal surfaces that are (1) geometrically good approximations, (2) topologically correct, and (3) consist of a small number of polygons. By topologically correct we mean that the connectivity of the constructed surface matches that of the true isosurface – any two points in the given sample are connected by a path that does not pierce the constructed surface, if and only if they are connected by a path that does not pierce the true isosurface.

We are interested in functions specified as the piecewise trilinear interpolant of a uniform mesh of sample points. The "marching cubes" algorithm of Cline et al. (1988) constructs a piecewise polygonal approximation to the isosurface, satisfying conditions (1) and (3) above, but not condition (2), i.e., the topology of the constructed surface may be incorrect. The "dividing cubes" algorithm of Cline et al. (1988) constructs a piecewise polygonal approximation to the isosurface satisfying conditions (1) and (2) above, but not condition (3), i.e., the constructed surface may not consist of a small number of polygons. Here, we present an efficient algorithm that constructs a piecewise polygonal approximation to the isosurface satisfying all three conditions, i.e., the constructed surface is geometrically a good approximation, topologically correct, and consists of a small number of polygons.

1 Introduction

Let $f(x, y, z)$ be a function from $[0, 1]^3$ to $[0, 1]$, where $[0, 1]$ is the unit interval on the reals. Let N be a natural number and let $\Delta = 1/(N-1)$. A *uniform sample* of f is a three-dimensional array F of N^3 points such that for i, j, and k in the range $0, 1, 2, \ldots, N\text{-}1$,

$$F(i,j,k) = f(i\Delta, j\Delta, k\Delta).$$

We call N the sampling number and Δ the sampling interval. The sample F partitions $[0, 1]^3$ into $(N-1)^3$ cubic *cells*.

We are interested in the problem of constructing a piecewise polygonal approximation S, to the isosurface I defined by $f(x, y, z) = t$, from a given sample F and a *threshold* value t.

The above problem is the abstraction of a data visualization problem that arises in several diverse areas, ranging from computer aided tomography in medical applications, to numerical simulations in fluid dynamics. See Fuchs et al. (1977), Hessellink and Wu (1988), Christiansen and Sederberg (1978), Artzy et al. (1980), Cline et al. (1988) and Wilhelms et al. (1990a). An alternative technique for visualizing such data is the technique of *volume rendering*, as discussed in Drebin et al. (1988), Levoy (1988) and Levoy (1991).

In visualization applications, the topology of the approximate isosurface is of considerable importance, as illustrated in the following scenario. A user employs isosurfaces for visualizing a sample, constructing piecewise polygonal approximations to the isosurfaces without preserving their topology. When examining a particular isosurface, the user notices an interesting topological feature, say, that the isosurface consists of two disconnected components. The curious user zooms in for a closer and more precise view of the surface, obtained by constructing an isosurface from a finer sample of the function. The disconnected components now appear connected, i.e., the topological feature vanishes. Such behavior is clearly undesirable, and will be eliminated by preserving the topology of the isosurface.

A certain practical issue must be addressed by an algorithm that constructs piecewise polygonal approximations to the isosurface; the number of polygons constructed must be small in order to minimize the time required to render the polygons on a graphics display device. It is reasonable to require that the number of polygons should be $O(M)$, where M is the number of the $(N-1)^3$ cells that intersect the isosurface $f(x, y, z) = t$.

Combining the above, we can formalize the problem as follows.

We say that two surfaces A and B are approximate within Δ, if for every point a on A there exists a point b on B such that b is contained in the cube of side 2Δ centered at a, and vice versa.

Isosurface Problem

input: A uniform sample F of an unknown function f over a sampling interval Δ, and threshold $t \in [0,1]$.

output: A piecewise polygonal surface S such that,

(a) S approximates the geometry of I, i.e., S and I are approximate within Δ, where I is the surface $f(x, y, z) = t$.

(b) S and I have the same topology, i.e., for every pair of sample points $u = (i_1\Delta, j_1\Delta, k_1\Delta)$, and $v = (i_2\Delta, j_2\Delta, k_2\Delta)$, u and v are connected by a path in $[0, 1]^3$ that does not pierce S if and only if u and v are connected by a path in $[0, 1]^3$ that does not pierce I.

(c) S is of low complexity, i.e., S consists of $O(M)$ polygons, where M is the number of the $(N-1)^3$ cells of F that intersect I.

It is clear that without restrictions on the nature of the unknown function f, the isosurface problem is ill-posed and possesses no algorithm. To remedy this, we can, for instance, limit the function f to be representable exactly by a known interpolant of the given data points. As it happens, the sampling number N is usually rather large, say 100, and the number of sample points is of the order of 10^6. Consequently, the use of smooth interpolants such as higher order polynomials or polynomial splines is forbiddingly expensive. (Wilhelms et al. (1990a) discuss the use of higher degree interpolants in isosurface construction.) Here, we settle for interpolation by the tensor product linear B-spline, more simply known as trilinear interpolation. The trilinear interpolant T interpolating the values of the function f at the vertices of a cube of side Δ may be expressed as follows:

$$T(x, y, z) = f(0,0,0) + [f(\Delta,0,0) - f(0,0,0)]x/\Delta +$$

$$[f(0,\Delta,0) - f(0,0,0)]y/\Delta + [f(0,\Delta,\Delta) - f(0,0,0)]z/\Delta +$$

$$[f(\Delta,\Delta,0) - f(0,\Delta,0) + f(\Delta,0,0) + f(0,0,0)]xy/\Delta^2 +$$

$$[f(0,\Delta,\Delta)-f(0,0,\Delta)+f(0,\Delta,0)+f(0,0,0)]yz/\Delta^2 +$$

$$[f(\Delta,0,\Delta)-f(0,0,\Delta)+f(\Delta,0,0)+f(0,0,0)]xz/\Delta^2 +$$

$$\Big[f(\Delta,\Delta,\Delta)-f(0,\Delta,\Delta)-f(\Delta,0,\Delta)+f(0,0,\Delta)\Big]xyz/\Delta^3 -$$

$$\Big[f(\Delta,\Delta,0)+f(0,\Delta,0)+f(\Delta,0,0)-f(0,0,0)\Big]xyz/\Delta^3.$$

Given a uniform sample of N^3 points, the function f can be represented as a *piecewise trilinear function* consisting of $(N-1)^3$ trilinear patches interpolating the given sample. Each patch is a surface in $\mathbf{R}^4$, and the entire collection of the $(N-1)^3$ patches is C^0 continuous. For a given threshold t, the isosurface $f(x,y,z) = t$ is simply the collection of the surfaces obtained by setting $T(x,y,z) = t$, for each of the $(N-1)^3$ trilinear patches T that compose f. Thus, when f is restricted to be the piecewise trilinear interpolant of the given sample, the isosurface $f(x,y,z) = t$ is easily defined. There remains the problem of constructing a piecewise polygonal approximation to the isosurface, satisfying the requirements (a), (b), and (c) listed under the definition of the isosurface problem.

As it happens, conditions (b) and (c) are difficult to achieve simultaneously, i.e., topological correctness is competitive with minimizing the number of polygons in the approximation. Cline et al. (1988) give two partial algorithms for the isosurface problem. Their first algorithm, the *marching cubes algorithm*, is simple, fast, and constructs $O(M)$ polygons. Unfortunately, the algorithm may produce isosurfaces that are not C^0 continuous and, therefore, can never be topologically correct. Their second algorithm, the *dividing cubes algorithm*, produces a large number of polygons, each roughly the size of the resolution of the display device. Within the resolution of the display device, the dividing cubes method eliminates the topological errors of the marching cubes method, but it does so at an enormous computational cost.

The marching cubes algorithm can be modified to eliminate the C^0 discontinuities using the methods of Wyvill et al. (1986) or Wilhelms et al. (1990a). (Also, in Appendix A, we suggest a simple modification of the marching cubes algorithm that achieves the same ends.) While these methods result in C^0 continuous surfaces, the topology of the surface may still be considerably different from that of the true isosurface.

In this paper, we present an algorithm for the isosurface problem for piecewise trilinear functions. The algorithm constructs a piecewise polygonal approximation with $O(M)$ polygons while guaranteeing topological correctness. In a practical implementation, we found that our algorithm was roughly 20% slower than the marching cube algorithm.

Formal proofs are omitted in this abstract, and are relegated to the full paper. Instead, we make heavy use of figures to explain the main ideas. To facilitate such, throughout this paper we use an *XYZ* coordinate system oriented as shown in Figure 1.

2 Topological Ambiguities

A trilinear interpolant is uniquely defined by the values of the function at the vertices of the cube. Thus, it is convenient for us to view the given sample of N^3 points as $(N\text{-}1)^3$ cubes, and examine the trilinear interpolants of each of these cubes. Consider a cube with function values specified at each vertex. The trilnear interpolant T varies linearly along each edge of the cube. Thus, for a given threshold t, linear interpolation will give us the points of intersection of the edges of the cube with the isosurface $T(x, y, z) = t$. Let us call these points the *edge points*. If we know how to connect up the edge points into triangles, then we will have a polygonal approximation to the isosurface. The manner in which the edge points are connected will determine the topological properties of the isosurface within the cube.

An isosurface will partition the vertices of the cube in such a manner that the vertices of the cube with function values above the threshold are isolated from the vertices of the cube with function values below the threshold. For convenience, let us say that vertices with function values greater than or equal to the threshold are black vertices, and vertices with function values below the threshold are white vertices.[1] Hence, the isosurface will isolate all the white vertices from the black vertices. In addition, it might also partition the white vertices into several disconnected subsets amongst themselves. Similarly, it might partition the black vertices into several disconnected subsets. In the following, we use the term "topology of the isosurface" loosely to refer to the above partioning of the vertices of the cube. For instance, Figure 2a shows a cube with two black vertices and six white vertices. Two distinct topologies are possible for such a coloring, and they are shown in Figure 2b.

Since there are eight vertices and each vertex may be colored white or black, there are 256 ways of coloring all the vertices. Owing to rotational symmetries, and the complementary symmetry of interchanging the white and the black colors, the 256 cases can be reduced to 15 base cases, Cline et al. (1988). These cases are shown in Fig. 3.

1. Throughout this paper, we lump "greater than" and "equal to" as one case. This simplification overlooks some degeneracies in the topology of the isosurface.

For a particular threshold value t, we say two vertices are *connected* if we can connect the two vertices by a path in the cube or on its boundary, such that the path does not intersect the isosurface $T(x, y, z) = t$. In the above, (a) if the path lies entirely on the boundary of the cube we say that the two vertices are *boundary connected*; and (b) if the path consists solely of edges of the cube, we say that the two vertices are *edge connected*; and (c) if the path lies entirely on a particular face of the cube, the two vertices are said to be connected on that face.

The following proposition is immediate.

Proposition 1: Two vertices are connected if they are edge connected.

Proposition 1 is sufficient to establish the topology of the isosurface for cases 0, 1, 2, 5, 8, 9, 11 and 14 of Figure 3. For the remaining cases, the topology is partially determined by Proposition 1. Figure 3 also shows the topology for the 15 cases as determined by the application of Proposition 1. The heavy edges in the figure designate edge connectivity.

Case 3 is one of the cases for which the topology is partially determined in Figure 3. Two distinct topologies are possible for this case, as depicted in Figure 2. We now outline a method for selecting between these topologies.

3 Saddle Points

The method is best understood in two dimensions. Consider the bilinear interpolant of four function values specified at the vertices of a unit square. The equation of such an interpolant is given by,

$$B(x, y) = f(0, 0) + [f(1, 0) - f(0, 0)]x + [f(0, 1) - f(0, 0)]y + \left[f(0, 0) + f(1, 1) - f(1, 0) - f(0, 1)\right]xy.$$

Suppose that we wish to construct the isocurve $B(x, y) = t$, for threshold t. As before, let us color the vertices white and black, with respect to the threshold t. There are three possible cases, taking into account rotational and complementary symmetries. These cases are shown in Figure 4a. The topology for cases 1 and 2 are unique, while for case 3, there are two possible topologies. For case 3, the bilinear interpolant will be as in Figure 4b, with a saddle point at (x_s, y_s). If $t > B(x_s, y_s)$, then the vertices with function values above t will be disconnected by the isocurves. If $t < B(x_s, y_s)$, then the vertices with function values below t will be disconnected by the isocurves. The case $t = B(x_s, y_s)$ is the degenerate form of either of the above cases. Thus, the topology for case 3 can be correctly determined by comparing the threshold with the value of the interpolant at the saddle point.

Similar to the saddle point of the bilinear interpolant, the trilinear interpolant has saddle points. The topology of the isosurface for a particular threshold can be determined by comparing the threshold value to the value of the interpolant at the saddle points. It is necessary to examine seven saddle points: one on each face of the cube, and one in the interior of the cube. The saddle points on the face are called *face saddles*, and the saddle point in the interior is called the *body saddle*. The values of the interpolant at the saddle points are called the *face saddle values* and the *body saddle values*, respectively. The calculation of the saddle points and the associated saddle values are treated in Appendix B.

Proposition 2: Two diagonally opposite vertices on a face of a cube are connected on that face if and only if at least one of the following holds.

(a) The two vertices are connected by a path of edges on the face.

(b) The function values at the two vertices, and the saddle value of the face, are all less than the threshold.

(c) The function values at the two vertices, and the saddle value of the face are all greater than or equal to the threshold.

With respect to a particular threshold t, let R be the set of all vertices of a cube with the same color, say white. We can decompose R into mutually disjoint *maximally boundary-connected* components R_1, R_2, etc, such that

(a) $R_1 \bigcup R_2 \cdots = R$.

(b) For each R_i, every pair of vertices in R_i are boundary connected.

(c) For all distinct pairs R_i and R_j, if $u \in R_i$ and $v \in R_j$, u and v are not boundary connected.

Proposition 3: Let R_1 and R_2 be two maximally boundary-connected components of a cube. R_1 and R_2 are connected if and only if there exists $u \in R_1$, $v \in R_2$ such that uv

is a body diagonal of the cube and one of the following holds.

(a) The function values at u and v, and the body-saddle value of the cube are all less than the threshold.

(b) The function values at u and v, and the body-saddle value of the cube are all greater than or equal to the threshold.

Using the above propositions, we can write the following algorithm for constructing a topologically correct piecewise polygonal approximation to the isosurface $T(x, y, z) = t$ within a cube.

Algorithm *saddle*
input:
    the function values at the vertices of a cube, threshold t;
output:
    topologically correct piecewise triangular approximation
    to the isosurface $T(x, y, z) = t$;
begin
    color vertices with function value greater than
    or equal to the threshold black;
    color vertices with function value less than the threshold white;
    Apply Proposition 1;
    if topology is not fully determined **then**
        compute face-saddle values;
        apply Proposition 2 to construct maximally
        boundary-connected components;
        if topology is not fully determined **then**
            compute body-saddle value;
            apply Proposition 3.
        end
    end
    compute edge-points;
    link edge-points to construct triangles consistent with topology;
    output triangles;
end

To illustrate the working of algorithm *saddle*, Figure 5 shows the computational tree obtained when *saddle* is run on case 6 of Figure 3.

We can now use algorithm *saddle* to construct isosurfaces for an entire sample F. To do so, it suffices to run *saddle* on each of the $(N-1)^3$ cubes composing F. This is wasteful, since only M cubes will contribute to the isosurface and M may be much smaller than $(N\text{-}1)^3$. More efficiently, it is easy to see that the $(N-1)^3$ cubes of F may be preprocessed into an octree, so that for a given threshold t it is possible to visit all M cubes that intersect the isosurface in time $M\log(N)$. (Wilhelms et al. (1990b) discuss the use of octrees at length and present some experimental data.) Algorithm *saddle* is run on each of the M cubes visited. Notice that *saddle* runs in constant time, producing no more than a constant number of triangles per cube. Also, the surface composed of the collection of these triangles for all of the M cubes will be C^0 continuous, except perhaps at the boundary of the data set. To see this, it suffices to note that the connectivities on a face are determined only by the function values on that face, and hence are common to both cubes sharing the face. For more on the sufficiency of this condition, see Wilhelms et al. (1990b).

4 Conclusion

We presented an efficient algorithm for the isosurface problem for the restricted case of piecewise trilinear functions. The surface constructed by the algorithm is topologically correct, piecewise polygonal, and is guaranteed to have a small number of polgyons.

Akin to trilinear interpolation over a cube, we have the class of trimonotonic interpolants, wherein the weighting functions are arbitrary monotonic functions rather than linear functions. It can be shown that our algorithm holds without modification for the more general case of piecewise trimonotonic functions.

Lastly, we note that the method of saddle points can be also be used for the determination of the topology of the isosurface of more complex interpolants such as quadrics and cubics. Unfortunately, the computations are prohibitively expensive.

5 Appendix A

Cline et al. (1988) give 15 base cases and the corresponding polygons to be used as a table for the marching cubes algorithm. Unfortunately, their algorithm produces surfaces with C^0 discontinuities or "holes", topological features that are inherently impossible in the isosurface of a continuous function. This was formally noted by Durst (1988). To eliminate these discontinuities, it suffices to follow a greedy algorithm, that resolves topological ambiguities in favor of connecting white vertices. The algorithm uses the 18 base cases of Figure 6.

Algorithm *greedy_cube*
input:
    the function values at the vertices of a cube, threshold t.
output:
    piecewise triangular approximation
    to the isosurface $T(x, y, z) = t$.
    output topology favors connecting vertices with
    function value less than threshold;
begin
    color vertices with function value greater than
    or equal to the threshold black;
    color vertices with function value less than the threshold white;
    check whether the coloring corresponds to one
    of the 18 cases of Figure 6;
    if not, complement the white and black colors and check again;
    compute the edge-points;
    construct and output the corresponding triangles;
end

We can now use algorithm *greedy_cube* to construct isosurfaces for an entire sample F. As mentioned earlier, the $(N-1)^3$ cubes of F may be preprocessed into an octree, so that for a given threshold t it is possible to visit all M cubes that intersect the isosurface in time $M\log(N)$. Algorithm *greedy_cube* is run on each of the M cubes visited. Notice that *greedy_cube* runs in constant time, producing no more than a constant number of triangles per cube. Also, the surface composed of the collection of these triangles for all of the M cubes will be C^0 continuous, except perhaps at the boundary of the data set. To see this, simply note that the connectivities on a face are determined only by the function values on that face, and hence are common to both cubes sharing the face.

6 Appendix B

We now sketch the calculation of the saddle points of the trilinear interpolant in a cube. For convenience, we write the trilinear interpolant T in the form:

$$T(x, y, z) = axyz + bxy + cyz + dzx + ex + fy + gz + h.$$

The face-saddles are not true saddle points of T, but are the saddle points of T limited to each face of the cube. For instance, for the face $x = 0$, T reduces to the bilinear interpolant on that face, and can be written

$$T(0,y,z) = cyz + fy + gz + h.$$

Setting

$$\frac{\partial T}{\partial y} = \frac{\partial T}{\partial z} = 0,$$

and solving, we find the saddle point at $(0, -f/c, -g/c)$, with a saddle-value of $h - fg/c$. Similarly, we can calculate the face-saddle values on the other five faces.

The body-saddle point is the solution to the system of equations

$$\frac{\partial T}{\partial x} = ayz + by + dz + e = 0.$$

$$\frac{\partial T}{\partial y} = axz + bx + cz + f = 0.$$

$$\frac{\partial T}{\partial z} = axy + cy + dx + g = 0.$$

If $a = 0$, the solution is given by:

$$x = \frac{ec - df - bg}{2bd}$$

$$y = \frac{-ec + df - bg}{2bc}$$

$$z = \frac{-ec - df + bg}{2cd}$$

The body-saddle value can then be obtained by substituting the above values for x, y, and z in the equation for $T(x, y, z)$.

If $a \neq 0$, we have

$$x = \frac{1}{a}\left[-c \pm \left[\frac{(bc - af)(ag - cd)}{(ae - bd)}\right]^{1/2}\right]$$

$$y = -\frac{(dx+g)}{(ax+c)}.$$

The body-saddle value can then be obtained by substituting the above values for x, y, and z in the equation for $T(x, y, z)$.

7 References

Artzy, E., Frieder, G., and Herman, G. (1980). The theory, design, implementation and evaluation of a three dimensional surface generation program, *Computer Graphics*, Vol. 14, No. 3, pages 2-9.

Christiansen, H. N., and Sederberg, T. W. (1978). Conversion of complex contour lines into polygonal element mosaics, *Computer Graphics*, Vol. 12, No. 3, pages 187-192.

Cline, H. E., Lorenson, W. E., Ludke, S., Crawford, C. R., and Teeter, B. C. (1988) Two algorithms for the reconstruction of surfaces from tomograms. *Medical Physics*, Vol. 15, No. 3, pages 320-337.

Drebin, R. A., Carpenter, L., and Hanrahan, P. (1988). Volume rendering, *Computer Graphics*, Vol. 12, No. 3, pages 187-192.

Durst, M.J. (1988). Letters: Additional reference to "marching cubes". *Computer Graphics*, Vol. 22, No. 2.

Fuchs, H., Kedem, Z. M., and Uselton, S. P. (1977). Optimal surface reconstruction from planar contours, Communications of the ACM, Vol10, pages 693-702.

Levoy, M. (1988). Display of surfaces from volume data, *IEEE Computer Graphics and Applications*, May, pp29-37.

Levoy, M. (1991). Viewing algorithms, in *Volume Visualization*, IEEE Computer Society Press, Los Alamitos, CA.

Wilhelms, J., and Van Gelder, A. (1990a). Topological considerations in isosurface generation. *Computer Graphics*, Vol. 24, No. 5, pages 79-86.

Wilhelms, J., and Van Gelder, A. (1990b). Octrees for faster isosurface generation. *Computer Graphics*, Vol. 24, No. 5, pages 57-62.

Wilhelms, J., and Van Gelder, A. (1990b). Octrees for faster isosurface generation. *Computer Graphics*, Vol. 24, No. 5, pages 57–62.

Wyvill, G., McPheeters, C., and Wyvill, B. (1986). Data structures for soft objects, *The Visual Computer*, Vol. 2, No. 4, pages 227–234.

Wu and Hessellink (1988), Computer display of reconstructed 3-D scalar data, *Applied Optics* Vol. 27, No. 2, pages 395–404.

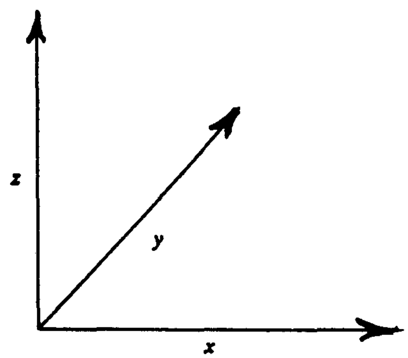


Figure 1: The coordinate system.

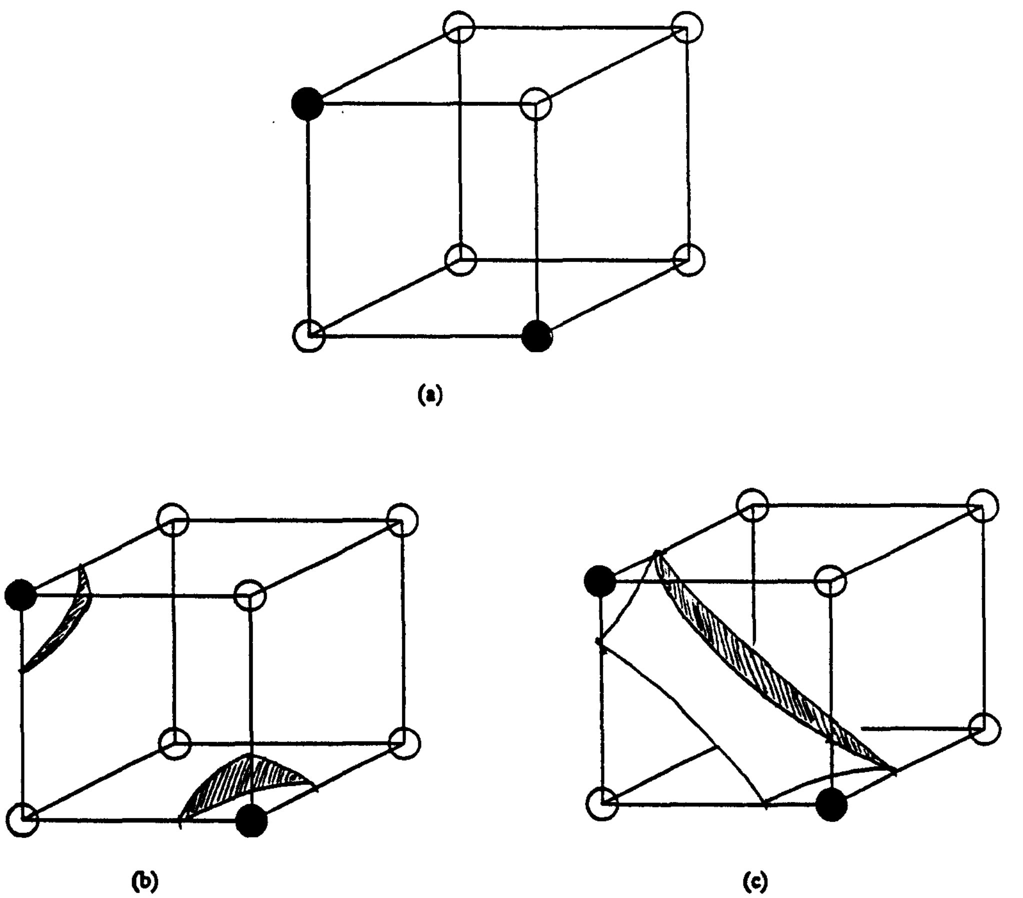


Figure 2: (a) Coloring a cube; (b) and (c) two topologically distinct isosurfaces for the coloring of (a).

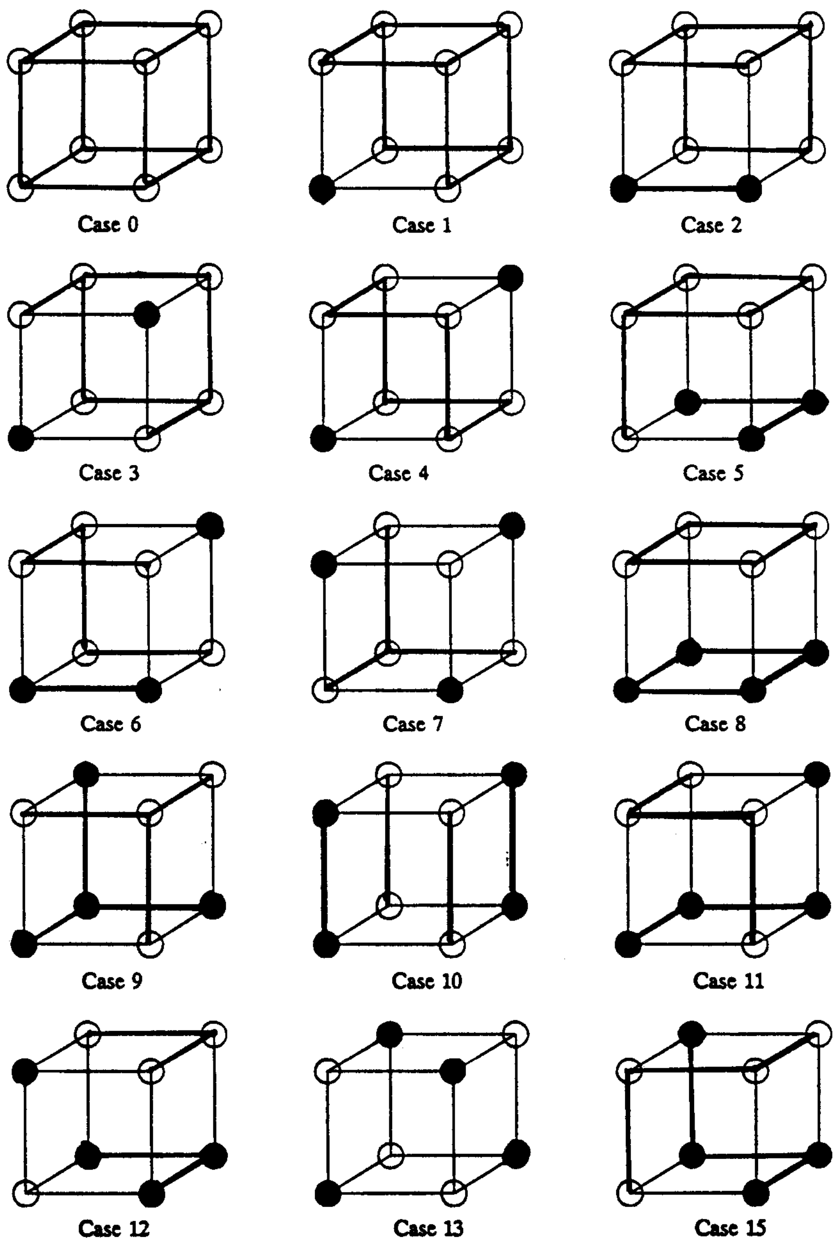
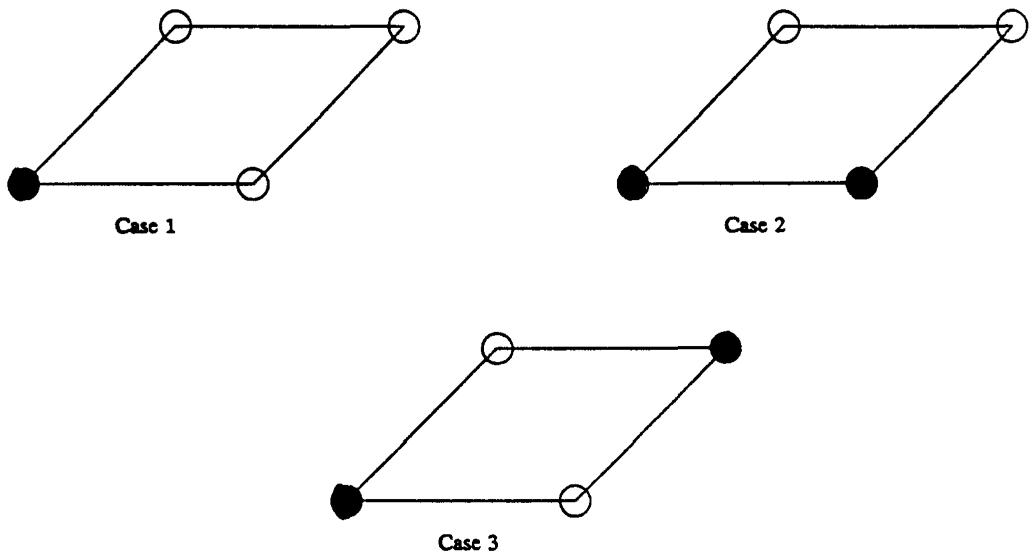
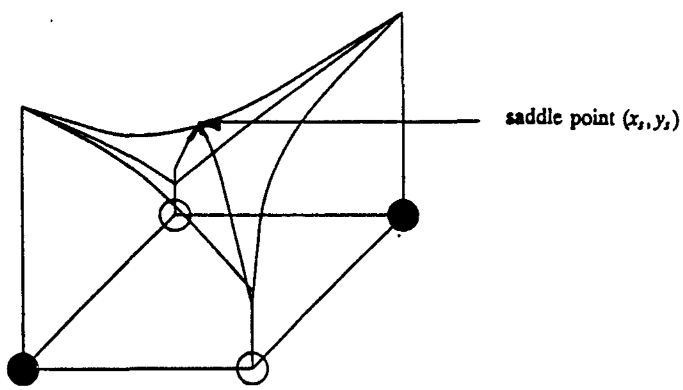


Figure 3: The 15 base case colorings for a cube. The heavy edges represent connectivity.



(a)



(b)

Figure 4: (a) The three base case colorings for a square. (b) The bilinear interpolant for the last of the cases in (a).

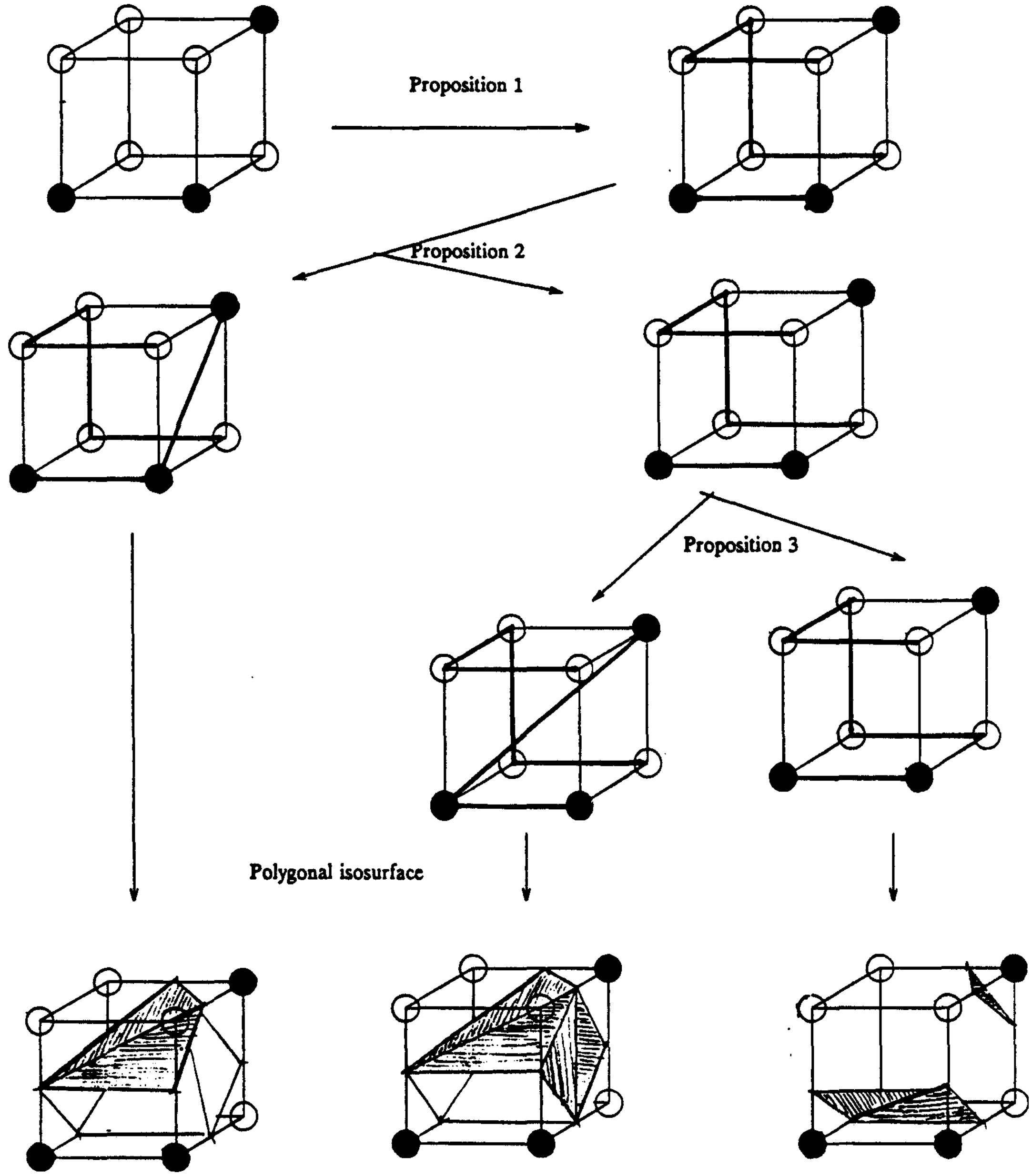
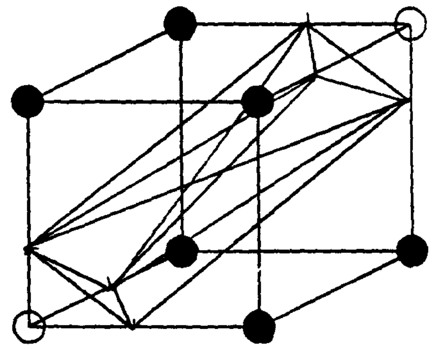
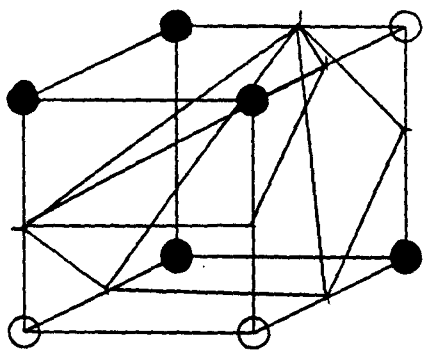


Figure 5: Applying algorithm *saddle*.

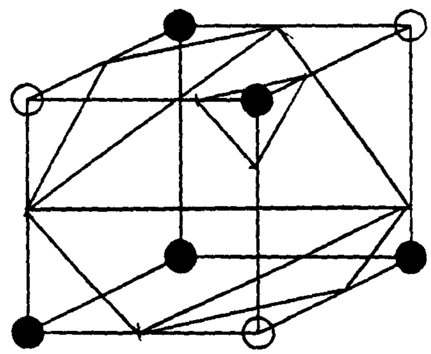
Cases 0-14 of Fig.1 of Cline et al. (1988)



Case 15



Case 16



Case 17

Figure 6: The 18 base cases for algorithm *greedy_cubes*.