Face Octrees. Involved Algorithms and applications

Pere Brunet

Report LSI-90-14

FACE OCTREES. INVOLVED ALGORITHMS AND APPLICATIONS

Pere BRUNET
Universitat Politecnica de Catalunya
Barcelona Spain

ABSTRACT

Volume models can be effectively represented by means of octree structures, that recursively subdivide the space into cubic octants. Classical octrees are well suited for objects with very complex surfaces whereas other schemes, such as extended octrees, are well adapted for the representation of polyhedra. The present paper introduces face octrees as an scheme for the representation of solids limited by smooth free form surfaces. The representation is discrete and approximated, through a tolerance specified by the user. Anyway, an exact canonic surface can be derived from the face octree representation. The spatial complexity of the model as a function of the geometric properties of the surface is discussed, together with its ability for the interrogation and realization of boolean operations.

1. INTRODUCTION

Geometric modelling must provide efficient tools for the representation and operation of 3D objects. A geometric modelling system consists of a set of utilities that operate on a non-ambiguous model of the object [Req80] and that allow object creation, shape editing, geometric transformations, interrogations, shape operations and rendering of the models. A data structure for the representation of objects is said to be unambiguous if every internal representation corresponds to a single object in the real world. Ambiguous systems cannot model correctly real objects, as it is not possible to interrogate representations which correspond to two or more real entities.

Although some schemes for the representation of surfaces based on algebraic surfaces have been proposed [Sed85], [Dah89] in the last years, most of the existing representation models for surfaces are parametric, and describe them through a set of continuously connected polynomial (or rational) patches, [BFK84]. The mathematical model for the individual patches is often based either on the Bezier approach or on B-splines. Continuity between patches is either parametric, like in B-splines, or geometric, which ensures the coincidence of the first geometric parameters: tangent plane, curvature. The topology of the mesh of patches has evolved in the last years from the simple rectangular mesh in tensor product surfaces, to the possibility of managing complex topologies with non-rectangular patches, such as triangular Bezier patches.

Surface models can describe either open surfaces or closed sets of patches that enclose a finite volume. Solid models, on the other hand, represent closed regions of the space in a unambiguous way. Both models have evolved independently [Req80], and whereas surface models are prepared to deal with free form surfaces of complex form, volume models are usually restricted to objects limited by plane or quadric algebraic surfaces (cylinders, cones, spheres, etc) that are well suited for the classification tests (inside - outside) of points against solids.

The two most well known schemes for the representation of solid objects are the boundary representation and the constructive solid geometry [Req80]. In boundary representations, the solids are represented by their boundary which is represented in turn by a disjoint set of faces bordered by one or more circular rings of edges which intersect in vertices. The model stores both geometric and topological information describing the connection between neighbour geometric elements. Constructive solid geometry, in turn, represents solids by ordered binary trees. In it, non terminal nodes can represent boolean operations like union, intersection and difference while terminal nodes can be instances of primitive solids located conveniently in the space. Both schemes have specific advantages and disadvantages: whereas boolean operations are computationally expensive in boundary representations, in CSG trees both rendering operations and geometric interrogations are rather complex. Moreover, both models are usually restricted to planar or quadric surfaces.

Octrees, being one of the classical decomposition schemes [Req80], [Mea82], are trees that represent solids by encoding the recursive subdivision of a finite universe. Classical octrees [Mea82], [SaW88] approximate the surface of the solid by a layer of small cubes of a minimum specified size. Although they are approximate representation schemes, they are well suited for the representation of objects limited by very complex surfaces, where the usual schemes fail (they can for example represent parts of the human body as the brain, lungs, etc). Some octree schemes have appeared for the specific representation of polyhedra, such as extended octrees [BrN85], [Nav86]; they are exact and very concise, but they are not suitable for the representation of solids limited by general surfaces. On the other hand, a generalization of extended octrees [BrA87], [Aya88] has been proposed for the specific representation of surfaces composed by biquadratic patches.

The present paper introduces face octrees, an intermediate scheme between classical and extended octrees which is appropiate for the representation of solids limited by free form surfaces. The representation is discrete and approximated, through a tolerance specified by the user. Anyway, an exact canonic surface can be derived from the face octree representation and thus, a surface model of the boundary of the object can be generated. Volume operations and interrogations can be performed through the octree structure, whereas some surface interrogations require the computation of the canonic surface. After a brief introduction on octree representations, section 2.3 includes the definition of face octrees. Section 2.4 discusses the discrete encoding of the associated planes, for the sake of robustness in the operations. Section 3 presents several algorithms for the generation of face octrees from other models, and section 4 discusses the spatial complexity of the proposed scheme as a function of geometric parameters in the solid being represented; face octrees complexity is discussed in front of the spatial complexity in the extended octree representation of a facetted approximation of the same solid. Finally, section 5 presents the basic ideas related to the computation of the canonic surface, boolean operations and interrogation of the model.

2. OCTREE REPRESENTATION OF SOLID MODELS

An octree is a tree that represents solid objects through the recursive subdivision of a finite cubic universe. In this structure, each tree node is terminal or has eight descendants. The tree divides the space of the universe into cubes of different sizes. The root of the tree represents the universe, a cube with a 2^N edge length. This cube is divided into eight identical cubes, called octants, with an edge length of 2^{N-1}. Each octant is represented by one of the eight descendants of the root. If an octant contains a too complex part of the solid (Grey node), it is divided into another eight identical cubes which are represented as descendants of the octant in question. The previous process is repeated recursively until valid terminal nodes are obtained, [Mea82]; the process also stops when octants of a minimum edge length called resolution (usually $2^0 = 1$) appear. Consequently, an specific octree representation is characterized by the definition of the types of terminal nodes which are allowed. The more extensive is the set of geometric cases considered in the definition of the allowed terminal nodes, the less intermediate (Grey) nodes in the octree representation of a particular object. On the other hand, the size and location of a cubic octant are determined by the level and the position within the octree of its associated node.

The next subsections will present some of the main octree representations which have been used for the representation of solids. After this, a new representation scheme called Face octrees will be introduced, in section 2.3.

2.1 Classical octrees

In the simplest octree representation, which we will call classical octrees (CO) in the rest of the paper, only Black and White terminal nodes are allowed. Both types of nodes are homogeneous terminal nodes. Nodes with associated cubic octants which are completely inside the object are coded as Black nodes, whereas those completely outside it are called White nodes, [Mea82].

Several proposals related to the effective encoding of octrees have appeared in the literature. The most important are the encoding as an explicit tree, [SaW88], the linear encoding of the black nodes of the tree, [Gar82], and the pre-order linear encoding or depth-first method (DF), [KaE80]. They could be used not only for classical octrees, but also for the other octrees that will be presented in the rest of this section.

2.2 Vector octrees. Extended octrees.

Octree models that incorporate new terminal nodes containing parts of the object surface can also be derived. They can be called vector octrees, [Sam89]. They present some advantages over classical octrees, as it can be shown [Nav86] that they can yield exact representations for polyhedra. On the other hand, they also reduce the degree of subdivision and therefore require less storage than classical octrees, [Nav86]. Different models can be considered, depending on the specific node types used in the representation. Restricting ourselves to the representation of polyhedral solids, Face, Edge and Vertex nodes [Nav86] can be used in adition to the classical White and Black terminal nodes, figure 1. Face nodes are crossed only by a single planar face of the solid, whereas Edge nodes contain two neighboring faces and a part of their common edge, and Vertex nodes contain one vertex of the polyhedron and part of the faces and edges converging to it, figure 1. In this context, several new octree models can be defined, [Nav86]. Face octrees are octrees in which the set of terminal nodes includes White, Black and Face nodes. If edge nodes are added to the face octree representation, the face-and-edge octree representation follows. Finally, an extended octree is obtained if vertex nodes are in turn incorporated [ABJN85], [BrN85] (A similar representation scheme, called Polytrees, was independently proposed in [CCV85]; see for instance [BrN90] for a discussion on the relative performances of both schemes). On the other hand, figure 2, in face octrees minimum size nodes are generated along the edges of the polyhedron , whereas in the face-and-edge octrees they appear near the vertices; extended octrees have no minimal size nodes, in general. However, deep subtrees can appear in the vicinity of vertices in extended octrees, depending on their specific location. Durst and Kunii [DuK89] refer to them as black holes, and in order to avoid them, they propose to allow nodes with faces that converge to an egde or to a vertex, in the Integrated Polytrees. The total number of nodes in the octree is then considerably reduced, [DuK89]. See also [Nav86] and [BrN90] for a discussion concerning relative performances and storage requirements of the extended octrees.

2.3 Face octrees

As it has been seen, Face octrees are octrees in which face nodes are allowed, in addition to the classical white, black and grey nodes. In general any type of faces (planar, quadratic, etc) can be allowed in face nodes, the only restriction being that precise information about the geometry of the face must be included in them. For instance, every face node can include either a pointer to the mathematical equation of the face or explicit geometric information about it. On the other hand, for reasons of simplicity only face nodes with linear geometry (plane faces) will be considered in the rest of the paper. The following definition precises the concept of face octrees:

Definition 2.1 A face octree is an octree which allows White, Black, Grey and Face nodes, together with a tolerance ϵ. White, black and grey nodes have the same meaning as in classical octrees. Face nodes contain part of the surface of the object, and they include in their encoding the geometric information about an associated oriented plane π_C, such that for every point P of the object surface S in the cube associated to the node, $dist(P, \pi_C) \leq \epsilon$, figure 3.

Consequently, the part of the surface of the object within a face node must be sufficiently planar. The value ϵ associated to the face octree controls the degree of approximation of the representation together with the depth of the tree. The encoding of face nodes must include both the corresponding type of node and geometric information on the associated plane. This can be either the explicit plane equation or a pointer to a table of face plane equations.

Face octrees can be used for the representation of polyhedra. In this case, the representation is more compact than the corresponding classical octree and exact, provided that every face of the polyhedron has at least one face node pointing to it. As the number of faces in the polyhedron is limited, it is in general better to have a pointer to the exact plane equation in every face node than to include the explicit geometric information in it. Algorithms can be derived, [Nav86] for the re-computation of the boundary representation of the object.

On the other hand, face octrees can also be used for the representation of objects limited by complex (free form) surfaces. As only plane faces are allowed, the octree representation is obviously approximate. However, the piece of surface represented in a face node depends only on its flatness; the artificial bondaries between smoothly connected patches in the surface model dissapear in the face octree representation. Decreasing the tolerance leads to deeper trees since previous terminal face nodes have to be subdivided. In this case, as the number of different associated planes in all face nodes can be very high, it is better to include the explicit equation of the associated plane in every face node. And, as it is only required that the plane approximates the surface within the node, a restricted discrete set of candidate planes can be used by limiting both the set of feasible normal directions and the distance to the origin. Next section, 2.4, introduces and discusses the encoding of associated planes.

Every face node in the face octree representation of a closed free form surface can be interpreted as a band which spans a distance equal to the tolerance at both sides of the plane associated to the node, and which is limited by two planes parallel to it. Because of the definition of a face node, the exact surface must lie in the band. Therefore, the whole boundary of the object is contained in the region defined by the set of bands of the face nodes in the octree, figure 4. Bands of neighbour nodes must overlap at the commom boundary, although not completely. The union of all face node bands defines a 'thick surface' with a width of twice the tolerance, which contains the true surface of the object. The whole surface can be analyzed by studying the properties of the coded thick surface, and no artificial geometric elements such as boundaries between smoothly connecting patches need to be considered.

As it will be shown in Section 6, a canonic surface can be obtained from the union of bands or thick surface associated to a face octree. This process can be interpreted as the recomputation of a canonic boundary representation from the face octree. The double conversion from a surface model to a face octree and back to the corresponding canonic surface generates the representative surface of the initial one. It can be concluded that, although the model is approximated, it can generate a precise geometric model for interrogation or rendering purposes.

2.4 Face octrees: discrete encoding of the associated planes

As it has already been stated, the encoding of face nodes in a Face octree representation must include both a code for their type and information on the associated plane. This last information can be encoded in any of the following three forms,

- A pointer to a table of oriented face plane equations. Although this scheme can be useful in the representation of polyhedra, it is not convenient while used for free form surfaces. The reason is that the surface will in general be approximated by a different plane in every face node. As a consequence, the use of the table of equations will not reduce the total number of face equations being stored.

- The explicit equation of the associated plane. In this case, the number of plane equations is the same as in the previous scheme; however, the amount of storage corresponding to the pointers is now saved.

- An index value that determines the associated plane within a finite set of feasible planes. The finite set of planes is obtained by discretizing both the normal vector direction and the distance to the origin, as it will be precised in the rest of the section. Every element of the finite set will be called a discrete plane.

Encoding of face nodes by discrete planes allows integer manipulation of face octree represented objects. Consequently, inherent imprecission problems related to floating point representations are avoided. This is essential for the robustness of boolean operation algorithms.

The overall surface of the object will lie in a set of bands of a bounded width, which can be represented by integer variables. Face octrees are a scheme that uniformly approximate the objects's surface, in the sense that the representation error is rather constant all over the surface. They inherit the property of uniform approximation from classical octrees, while being more compact and storing more information on the surface shape. On the other hand for instance, boundary representations try to represent exactly the equation of the faces of the surface, being however approximate [Hof89] because of the floating point representation of the plane equations.

A possible representation for discrete planes follows. It is based on a finite-bit precission encoding of both the normal vector and the distance to the origin d of the plane equation $ax + by + cz + d = 0$. In particular, our proposal uses $2n_n + n_d + 4$ bits and includes,

- Three bits for the encoding of the sign of the three components of the normal vector (a, b, c). In other words, they encode the octant pointed by this normal vector.

- n_n bits which encode the normalized value $\frac{|a|}{|a|+|b|+|c|}$ with finite precission between 0 and 1.

- n_n bits that encode $\frac{|b|}{|a|+|b|+|c|}$ in a similar way.

- One bit for the encoding of the sign of the last value d of the equation.

- n_d bits which encode the magnitude $\frac{|d|}{\sqrt{3}\,2^N}$ with finite precission between 0 and 1 ($\sqrt{3}\,2^N$ is the length of the diagonal of the universe cube, being the maximum value of $|d|$).

Encoding a particular plane equation with this scheme is obviously immediate. The inverse process of decodification must only compute $|c|/(|a| + |b| + |c|)$ as one minus the sum of the two encoded normalized components of the normal vector.

Figure 5 presents a geometrical interpretation of the encoding of $a_n = |a|/(|a| + |b| + |c|)$ and $b_n = |b|/(|a| + |b| + |c|)$ with finite bit precission. They can be interpreted as barycentric coordinates of an equidistant mesh of points in a triangle that bisects the corresponding octant around the origin, figure 5-a. Representable normal vectors are those pointing to particular points in the mesh, figure 5-b. The mesh subdivides every side of the triangle in 2^{n_n} sub-triangles. The whole set of points in the mesh is representable except two of the vertices of the main triangle, because $a_n < 1$ and $b_n < 1$. However, this is not a major problem, as a unique representation is needed for the coordinate directions and in general for directions in the coordinate planes. In fact, every direction pointing to an edge between two triangles in different octants must belong unambiguously to one of them; in the same way, every one of the six coordinate directions belongs to a specific octant, figure 5-c and is represented in an exact way.

Concerning normal vectors, this discrete representation scheme has two basic problems. The first one is anisotropy: because of the rough approximation of the unit sphere by a octahedron, some directions are better represented than others: the maximum representation error, defined as the angle between the normal vector pointing to the barycenter of the central triangle in the mesh and the closest representable normal vector, can be shown to be $\sqrt{2}\,2^{-n_n}$, whereas the minimum one, corresponding to the coordinate directions, is $(\sqrt{2}\,2^{-n_n})/3$. On the other hand, because of the restrictions on the barycentric coordinates that must sum up to one, up to $2n_n$ bits are needed to represent a set of $((2^{n_n} + 1)\,(2^{n_n} + 2))/2 - 2$ points in the mesh in every triangle. The first problem could be avoided by using a triangular mesh more adapted to the unit sphere (e.g., the central projection on the sphere of the subdivision of a regular icosaedron). In order to avoid the second problem, a recursive subdivision in four parts - by halving every edge in every step - of the initial triangle could be used, together by a quadtree-like representation of the specific subdivision process that leads to the mesh triangle containing the normal vector being encoded. Although both schemes could be investigated, we do not use them in our representation scheme, the reason being that they increase significatively the encoding and decoding time for normal vectors.

We have already pointed out that the maximum angular error in the representation of normal vectors is found in the bisector directions of the octants and is,

$$\sqrt{2}\,2^{-n_n}$$

On the other hand, the minimum distance between two distinct parallel representable planes gives us the other measure of the representation error,

$$\sqrt{3}\,2^N\,2^{-n_d}$$

Let us suppose now that the surface of the object being encoded is continuous, and note by ϵ_t the maximum distance, for every point P of the surface, between the tangent plane π_P to the surface in P and the surface itself at a distance $\sqrt{3}$ from P (see the definition of $d\_eps(P)$ in section 4). Then, the whole surface can be encoded in face nodes using the proposed representation for discrete planes and a tolerance ε such that,

$$\varepsilon \leq \epsilon_t + \frac{\sqrt{3}}{2} 2^N \left(\frac{\sqrt{2}}{2} 2^{-n_n} + 2^{-n_d}\right)$$

In order to prove this inequality, we must first observe that for any node containing part of the surface of the object, it always can be subdivided into valid face nodes with edge length ≥ 1, using a tolerance ϵ_t and no discrete representation for planes. This follows immediately from the above definition of ϵ_t. Now, once the surface has been encoded into theoretical bands with tolerance ϵ_t, these bands can be represented, in a second step, using the finite set of discrete planes. This representation process causes an enlargement of the band, from ϵ_t to a value not larger than ε. Assuming the worst angular position for the theoretical band, the angle between the band and its corresponding discrete plane will be half the maximum representation error, $\frac{\sqrt{2}}{2} 2^{-n_n}$, and the maximum enlargement of the band will appear in the largest face nodes of the octree. Assuming that face nodes can be as large as the univers cube, with edge length 2^N, the enlargement of the band produced by this concept is $\frac{\sqrt{3}}{2} 2^N (\frac{\sqrt{2}}{2} 2^{-n_n})$, which stands for the first term in the above espression for ε. Independently, the worst spatial location for the theoretical band is the one equidistant between two distinct parallel representable planes; in this case, the corresponding enlargement of the band is $\frac{\sqrt{3}}{2} 2^N 2^{-n_d}$, which stands for the second term in the expression for ε.

As a corollary, we can say that the representation of planes using a discrete encoding enlarges the required bands in face nodes. However, being given a tolerance ε, the representation error can be made as small as possible (and ϵ_t similar to ε) by sufficiently increasing the number of bits n_n, n_d in the representation scheme, and a bound of the difference between ε and ϵ_t has been obtained. Then, it can be concluded that the proposed scheme of discrete planes is a valid representation for face nodes.

3. GENERATION OF FACE OCTREES

Suppose we have an object with a smooth surface, and we want to build a Face octree to represent it. The steps to follow are homologous to building a classical octree, except that an extra consideration must be made when a node is not empty nor full. In fact, the process could be described by the following recursive algorithm that generates the octree in preorder,

```
procedure build_FO (list_faces,x,y,z,scale)
          clipping (list_faces,x,y,z,scale,list_real_faces)
          if  no faces and  node_is_outside then
                write_node (White)
          elseif  no faces and  node_is_inside then
                write_node (Black)
          elseif  flat_enough then
                Compute_discrete_plane; write_node (Face)
          else
              write_node (Grey)
              foreach of the eight subnodes  do
                 compute_node_coordinates (xi,yi,zi)
                 build_FO (list_real_faces,xi,yi,zi,scale/2)
              enddo
          endif
end_procedure
```

The two main parts in the algorithm are the clipping procedure, that generates the list of real faces (or patches) of the surface that intersect the cube associated to the octree node, and the function that detects if the surface is flat enough and can be approximated to within ϵ by a plane of the discrete set. The reduced list of faces in list_real_faces is then sent to the nodes in the subtree, if the node is found to be grey.

In order to determine if a piece of boundary surface contained in the node is flat enough or not, the specific algorithm depends on the actual representation of the surface. For Bezier patches, for instance, existing procedures for the computation of bounding boxes can be used, provided that they are generalized to handle groups of patches in the list of real faces corresponding to the node.

On the other hand, solids with a smooth boundary are often described, for instance, by means of a collection of parametric patches that cover the boundary. The problem of determining whether a portion of space is completely contained in such an object is a hard one. In the context of building a face octree, however, this difficulty is diminished by the fact that when considering a certain cube, it must be true that the father node contained a portion of the boundary of the solid. Therefore, the classification of the nodes that do not interfere with the boundary can be stablished based on the nodes that contain portions of the boundary, provided that a convention assuming outward normal in the patches is adopted. In a first step, an octree containing face nodes and undetermined black/white nodes can be built; then, the classification in interior and exterior in face nodes can be propagated in a germ-like manner to the rest of White-Black nodes of the octree. This could in fact be a way of testing whether a set of patches bounds a well-defined solid in a not too expensive way.

The presented algorithm could also be used for the face octree representation of objects with non-smooth surface, or even for polyhedra. In these cases, the function flat_enough should return false in most of the cases where the cube associated to the node contains edges or vertices of the solid (unless the overall surface inside the cube is sufficiently planar). As a consequence, the face octree representation of objects with non-smooth surface forces a high subdivision near sharp edges and vertices. See section 4 for a detailed discussion of the spatial complexity of the face octree representation.

The face octree of an object can also be obtained from its corresponding classical octree, by a simplification process: any grey node of the octree which models a sufficiently flat part of the surface of the object, can be substituted by the corresponding face node. The main idea in the conversion algorithm is to test every grey node of the initial octree in a preorder traversal: if all boundary nodes in the subtree lie in a band between parallel planes with a width twice the tolerance ϵ, then the node and its subtree can be substituted for a face node. In this algorithm, a boundary node is defined, for instance, as a black node which has at least one neighbour which is a white node. The only restriction to the algorithm is that the tolerance is bounded by d/2, d being the length of the diagonal of minimum-size nodes.

4. SPATIAL COMPLEXITY OF THE OCTREE REPRESENTATIONS

Defining the size of an octree as the total number of nodes that it contains, it is known that most of the algorithms related with octree representations show linear complexity with respect to the size of the octrees being operated. On the other hand, the same object can be coded in different octree representations, figure 6, and the size of the resulting octree in every of them depends on the geometric features of its surface. This section discusses the spatial complexity, in terms of the size of the octree, of three of the octree representations that have been introduced in the previous sections: the classical octrees (CO), face octrees (FO) and extended octrees (EO). Three parameters which measure the main related features of the objects's surface are introduced, and depending on the surface area S of the object and these parameters, it is shown that the optimal octree representation in terms of the efficiency of the algorithms can be different for every object, depending on its surface features.

Figure 6 shows the generation of the different octree representations (CO, FO, EO) for a general solid, together with the corresponding degrees of approximation. By recursively clipping the surface of the object against the tree nodes along the subdivision process, the CO representation can be generated. This model is obviously not exact, the degree of approximation being related to the size of the edges of minimum scale nodes. On the other hand, the FO representation can be derived if a tolerance ϵ is specified. This tolerance controls the degree of approximation of the face octree. Finally, the EO representation can be generated in an exact way from polyhedral models. However, when the initial solid is not a polyhedron, it must be approximated by means of a facetting operation. In this case, the model is obviously also approximated, not because of the EO model but of the facetted approximation. It must be observed that the scheme is also valid for polyhedral solids; now the EO representation is exact, provided that the solid model remains invariant through the facetting operation.

The aim of the rest of the section is to discuss the spatial complexity (size) of the different octree representations (CO, FO, EO) of the same object, provided that the same tolerance ε is used in all conversion processes, figure 6.

In general, classical octrees reach the maximum level of subdivision all over the surface of the object, which is therefore approximated by a staircase-like set of minimum scale nodes. As an immediate consequence of this fact, it can be shown [Sam89] that the size of a classical octree is of the order of the surface area S of the object.

On the other hand, the size of the FO and EO representations depends on the features of the surface of the object in a more complex way. Let us introduce the point functions d_eps, d_sph and d_face, which are defined for every point P of the object surface S as,

d_eps(P) = min (dist (P,Q)), for all points Q $\in$ S such that $dist(Q, \pi_P) \geq \varepsilon$, ε being the tolerance of the FO representation and π_P the tangent plane to S in P.

d_sph(P) is the radius of the maximum sphere centered in P such that the part of S inside the sphere is homeomorphic to a disc.

d_face(P) = min (dist(P,Q)), for all points Q $\in$ S such that $face(P) \cap face(Q) = \emptyset$. This function is only defined for polyhedra (figure 6), and therefore the faces face(P) and face(Q) that contain the points P and Q are assumed to be planar.

On the other hand, let us note by C(Q,n) the cube in R^3 which has vertices $(x + i * 2^n, y + j * 2^n, z + k * 2^n)$, i=0,1, j=0,1, k=0,1, (x,y,z) being the cartesian coordinates of the point Q; diag(n) will note the length of the diagonal of C, $diag(n) = \sqrt{3}\, 2^n$. The following propositions derive from the definitions of the point functions d_eps, d_sph and d_face,

Proposition 4.1. Given a certain cube C(Q,n) that contains part of the surface S of the solid, if for every point P in C such that $P \in S$ the following inequality holds,

$$min(d\_eps(P), d\_sph(P)) \geq diag(n)$$

Then the cube C(Q,n) can be a valid terminal node in the FO representation of the object.

In order to prove it, it must be observed that both d_eps(P) and d_sph(P) are not less than the length of the diagonal of the cube. The fact that $d\_sph(P) \geq diag(n)$ means that $S \cap C$ is homeomorphic to a disc centered in P. And this single sheet of surface S within C is sufficiently flat, as for any point R in C, R$\in$S, $dist(P, R) \leq diag(n) \leq d\_eps(P)$ and consequently $dist(R, \pi_P) \leq \varepsilon$. Indeed, the tangent plane π_P at any point P in C such that $P \in S$ can be used as the plane coding the face octree node.

Proposition 4.2. Given a certain cube C(Q,n) that contains part of the surface S of a polyhedral solid, if for every point P in C such that $P \in S$ the following inequality holds,

$$d\_face(P) \geq diag(n)$$

Then the cube C(Q,n) can be a valid terminal node in the EO representation of the object.

Let us note by F the set $\{face_1....face_k\}$ of faces totally or partially included in the cube C. In order to prove this proposition, it must be first observed that if the cube C contains a certain point P of the polyhedron, then the set of faces F can only contain face(P) and its neighbour faces through edges and vertices of face(P). In other words, it is not possible to find two faces $face_i$, $face_j$ in F such that $face_i \cap face_j = \emptyset$. As an immediate consequence, no more than one vertex of the polyhedron is in C. Two possible cases may arise,

- The cube C contains one vertex of the polyhedron. In this case, as the set F cannot contain faces not converging to this vertex, the cube is a vertex node in the EO representation.

- The cube C contains no vertex of the polyhedron. If the faces in F converge to a vertex outside C, then the node is a nearly-vertex node of the EO. On the other hand, if they do not converge to a vertex, the cube can contain zero, one or more edges of the polyhedron. In the first cases, the node is a face or edge terminal node in the EO representation; and the last case is impossible, since the existence of two edges in C not converging to a vertex of the polyhedron would imply the existence of two faces $face_i$, $face_j$ in F such that $face_i \cap face_j \neq \emptyset$, and this negates our previous statement. This completes the prove of proposition 4.2.

Let us suppose now that the surface map of the functions min(d_eps(P), d_sph(P)) or d_face(P) (figure 6) on S is known. This means that the zones of the surface with different values of the point function have been delimitated. Both surface maps can be generated a priori from the geometric model of the object, and they reflect local geometric features (curvature, thickness, size of the faces) of it. Figure 7 shows an example of surface map of the function d_face(P) in the case of a simple polyhedron.

Propositions 4.1 and 4.2 can be stated in the following way: if a cube C(Q,n) intersects a zone of S in which the value of the map of min(d_eps(P), d_sph(P)) is greater than diag(n), then C is a valid terminal node for the FO representation of the object. Similarly, if it intersects a zone of S in which the map of d_face(P) is greater than diag(n), then C is a valid terminal node for the EO representation of this object. In fact, subdivision of the octree is forced in the zones where the map presents small values, whereas big terminal nodes are allowed in the zones with high values of the map. More precisely, let us define by S_k^F and P_k^F the surface area and perimeter of the zone z_k of the surface of the object where

$$min(d\_eps(P), d\_sph(P)) \geq \sqrt{3}\, 2^{N-k-1} \qquad k = 0, 1, 2.., N-1$$

and

$$min(d\_eps(P), d\_sph(P)) < \sqrt{3}\, 2^{N-k} \qquad k = 1, 2.., N$$

(Obviously, the magnitudes S_k^F sum up to the total surface area of the solid). Then, the number of nodes in the FO representation of the object is bounded by a linear combination of the values S_k^F and P_k^F, as stated in Proposition 4.3:

Proposition 4.3. The total number of nodes in the FO representation of an object is bounded by the expression,

$$n_{nod}(FO) \leq 9 \sum_k (4^{k-N}\, S_k^F + 2^{k-N}\, P_k^F)$$

In order to prove it, let us note by n_i the number of face nodes of a particular size (edge length 2^{N-i}). The set of face nodes in the FO representation cover the whole surface of the solid. Now, following proposition 4.1, a zone z_k of the surface is covered by face nodes of edge length $\geq 2^{N-k}$. In the worst case, we can assume that all nodes covering z_k are the minimum allowed size, 2^{N-k}. Considering the minimum cross-section of these nodes, 4^{N-k}, as they spread all over z_k and along its perimeter, we can write the following expression for the total number of face nodes,

$$n_{face} = \sum_k n_k \leq \sum_k (4^{k-N}\ S_k^F + 2^{k-N}\ P_k^F)$$

On the other hand, the total number of black and white nodes n_{bw} can be bounded in terms of the number of face nodes, $n_{bw} \leq (55/8)\ n_{face}$. This expression follows by considering the worst case, in which all face nodes are at the deepest level of the tree and intermediate grey nodes have one grey and seven black/white son nodes. In this case, at the deepest level the number of black/white nodes is $\leq 6\ n_{face}$ (the subdivision of a node is only feasible if the surface spreads over more than one of tis sons). In addition, the number of black/white nodes not at the deepest level is $\leq 7/8$ of the total of nodes at the corresponding levels, which is the same as $\leq 1/8$ of the terminal nodes at the deepest level. Then,

$$n_{bw} \leq 6\ n_{face} + \frac{1}{8}(6+1)\ n_{face} = \frac{55}{8}\ n_{face}$$

Now, as it is well known [Sam89], [BrN90] the number of nodes in an octree fulfils $n_{leaves} = 7 * n_{grey} + 1$. Then, $n_{nod} = \frac{(n_{leaves}-1)}{7} + n_{leaves} \leq \frac{8}{7} n_{leaves}$. Proposition 4.3 follows now by considering that leaves include both face nodes and white and black nodes.

In the same way, if we define by S_k^E and P_k^E the surface area and perimeter of the zone z_k of the surface of a polyhedron where

$$d\_face(P) \geq \sqrt{3}\ 2^{N-k-1} \qquad k = 0, 1, 2.., N-1$$

and

$$d\_face(P) < \sqrt{3}\ 2^{N-k} \qquad k = 1, 2.., N$$

Then, the number of nodes in the EO representation of the object is bounded by a linear combination of the values S_k^E and P_k^E, as states Proposition 4.4:

Proposition 4.4. The total number of nodes in the EO representation of a polyhedron is bounded by the expression,

$$n_{nod}(EO) \leq 9 \sum_k (4^{k-N}\ S_k^E + 2^{k-N}\ P_k^E)$$

Proposition 4.4 can be proved in a completely parallel way to that of proposition 4.3; in this case, a first bound of the number of extended nodes (face, edge, vertex and nearly_vertex) is computed in terms of S_k^E and S_k^E.

Both propositions 4.3 and 4.4 tell us that the greater the surface areas S_k^F (S_k^E) for small values of k are (figure 7), the more compact the FO (E0) representation of the solid can be, as the subdivision process can stop very early at these zones. This is exactly the opposite of the classical octrees case, where the subdivision process must always continue since unit edge length nodes are found (the CO representation of a solid can be interpreted in terms of propositions 4.3 and 4.4 through considering $S_k^E = 0 \quad \forall k < N$).

Propositions 4.3 and 4.4 can also be used if only partial information about the surface maps of the magnitudes is available, by using lower bounds of the surface areas S_k^F, S_k^E in them. In this case however, the results are obviously less strong.

On the other hand, propositions 4.1 and 4.2 allow a local discussion on the complexity of the FO and EO representations. If inside any grey node of the octree having an associated cube, C(Q,n),$min(d\_eps(P), d\_sph(P)) < diag(n) \leq d\_face(P)$, then this node is a terminal node in the FO representation, while a subdivision can be necessary in the EO representation. Conversely, the node will be a leave of the EO representation if $d\_face(P) < diag(n) \leq min(d\_eps(P), d\_sph(P))$. In the first case, the local spatial complexity of the EO is $n_{nod}(EO) \geq n_{nod}(FO)$ whereas in the second, $n_{nod}(FO) \geq n_{nod}(EO)$. As a limit case, we can consider objects with nearly constant curvature. In an infinite cylinder, for instance, it is easy to see that $d\_eps(P) = d\_face(P)$ if a facetted approximation is obtained following the scheme in figure 6. Then, it can be concluded that in this case both representations have a similar number of nodes (although the FO is more compact and efficient because of the simpler representation of face nodes). On the other hand, sharp corners with high curvature in the surface are well approximated by vertex nodes in the EO representation but force subdivision in the FO, the limit case being edges and vertices in polyhedral solids.

5. OPERATION AND INTERROGATION OF FACE OCTREES

Face octrees (figure 8) can be used for performing boolean operations, computing the result of geometric interrogations and obtaining the corresponding canonic surface. The present section presents the main principles in which the corresponding algorithms are based.

5.1 Generation of the canonic surface of a face octree

For some purposes, a boundary model of the solid may be needed. For instance, we may need to generate it in order to exchange information with other packages, or for rendering purposes. One possibility is to keep it along all the way; in this case, the FO acts more as an auxiliary model that aids in certain computations, but the original boundary model is available and therefore need not to be rebuilt. On the other hand, if the FO representation keeps no reference to the original surface, a canonic surface must be generated from the octree. The canonic surface of a FO can be interpreted as the representative of all the surfaces that generate the same FO because of lying within its set of bands.

One possible algorithm for the computation of the canonic surface [PIV90] is based on the degree elevation of the planes in face nodes. These planes can be considered as algebraic surfaces, controlled by weights in the vertices of the cubes associated to the nodes. Enlargement of the cubic domain together with the degree elevation of these surfaces leads to coincident domains for neighbour planes. The canonic surface is then obtained [PIV90] from weight averaging while preserving the tolerance of the bands and the location and normal vectors of the significative points. Significative points are the intersections of the canonic surface with the edges of the mesh of cubes associated to the FO nodes, and they are univocally determined - location and tangent plane - in the first step of the algorithm from the set of bands intersecting each edge of the mesh.

Among the applications of the canonic surface of the FO we can quote the conversion to a boundary representation, refinement of the FO in order to obtain an FO representation with a smaller tolerance, local refinement for boolean operation purposes (see section 5.2), precise geometric interrogations (see section 5.3) and rendering. Rendering requires both the computation of precise profiles from the point of view and the determination of normal vectors. These can be computed either by a 'Gouraud style' averaging of the normals at the significative points of the canonic surface, or by computing the canonic surface itself.

5.2 Boolean operations. General extended octrees

If the result of a boolean operation between FO models has to be coded in turn as a FO, a high degree of subdivision is required along edges and vertices of the resulting object, in order to fulfil the requirement of the ε approximation, figure 2. An alternative is to code the result of boolean operations as general extended octrees.

General extended octrees are a generalization of both FO and EO schemes, in the sense that both FO and EO representations are particular cases of general extended octrees. The same node types as in EO are allowed (figure 1), the only difference being that extended nodes (face, edge, vertex, nearly_vertex) substitute the pointers to the corresponding planes that exist in the EO representation [BrN85], [BrN90] by generalized pointers. A generalized pointer is a case record containing a flag and either an actual pointer to a face plane equation or a discrete representation of a plane, depending on the value of the flag. Thus, an FO is obviously a general extended octree that contains only face nodes of discrete type. It must be observed that a generalization of the face-and-edge octree (figure 2) could be also a valid representation of the result of boolean operations, the only difference being the maximum subdivision near the vertices of the final solid.

Both boolean operations between EO and FO and between FO and FO produce general extended octrees as a result. And the corresponding boolean operation algorithms are completely similar to those existing for extended octrees [Nav86], [BrN90], as their edge, vertex and nearly_vertex nodes are computed from the location of the intersections between the specific central planes defining their bands. The algorithms are based on a join preorder traversal of both trees, while operating homologous nodes [BrN90] and producing extended nodes from the intersection of planes of edge, vertex and nearly_vertex nodes in the initial trees. However, edges and vertices in general extended octrees are not exact but define precise zones of the space (prisms and small polyhedra) that are fuzzy in some sense, as they contain or may contain an edge or vertex of the final solid. The size of these zones and the degree of uncertainty depends on the angle between the faces being operated. If the two bands in the operating face nodes cross themselves, then the prism centered in the edge of the final edge node will contain an edge of the solid. However, nothing can be concluded if they intersect without crossing themselves. In this case, two solutions can be implemented: either the intersecting nodes are locally refined by using local computations of the canonic surface, or a postprocess of the boolean operations is performed in order to eliminate dangling bands and ensure that every band in the final solid is part of a boundary that encloses a region of the space.

5.3 Interrogation of face octrees

Face octrees can be used for volume and geometric interrogations. Fast algorithms exist for most of them if only an approximate answer is required, although more accurate answers may need the computation of the canonic surface. Some possible interrogations include,

- Volume computations. The enclosed volume, together with momentums of inertia and other volume properties can be easely computed in an approximated way from the contribution of the cubic black nodes and the inside part of face nodes. The exact contribution of the bands in face nodes requires however the knowledge of the canonic surface.

- Intersection between a straight line and the solid. The question whether the line intersects or not the solid can be answered in most cases without computing the canonic surface. Once the list of nodes crossed by the straight line has been built, the line can only intersect the solid if there is any face node in the list and it intersects the band of this node; moreover, if the line intersects both planes of the band, then it intersects the surface of the solid. The canonic surface must be computed in the case that the line intersects only one of the planes of the band; it must also be computed if either the exact coordinates of the intersection point or the tangent plane are required.

- Plane sections. Plane sections of a solid represented by means of a FO can be obtained by sectioning face nodes. The set of 2D bands in the section plane (figure 4) can then be approximated by a polygon joining significative points in this plane, or can be refined by using the concepts of the canonic surface.

- Point-solid classification. Face octrees provide a fast tool for the classification of points, asking if they are inside, on or outside the solid. The point coordinates must be tested against the cubes corresponding to the grey nodes in the tree traversing it from the root to the leaves and looking for the terminal node containing it. Obviously, the classification is inside if the node is black or the point is in the 'black part' of a face node. The only case which requires further computations (of the canonic surface) is when the point is inside a band in a face node. In this case, the problem can be solved by computing the intersection of a straight line with the band, and comparing both the given and the intersection point. The result can obviously be any of the three, in this case.

6. CONCLUSIONS

The present paper has introduced face octrees, an intermediate scheme between classical and extended octrees which is appropiate for the representation of solids limited by free form surfaces. The representation is discrete and approximated, through a tolerance specified by the user. Anyway, an exact canonic surface can be derived from the face octree representation and thus, a surface model of the boundary of the object can be generated. Volume operations and interrogations can be performed through the octree structure, whereas some surface interrogations require the computation of the canonic surface. High level algorithms for the generation of face octrees from other models have been presented. The spatial complexity of the proposed scheme as a function of geometric parameters in the solid being modelled has been discussed.

Future work includes the investigation of efficient algorithms for the computation of the canonic surface and boolean operation of general extended octrees, together with the analysis of the robustness of the algorithms through the discrete representation of planes.

7. ACKNOWLEDGEMENTS

I would like to sincerely thank Isabel Navazo and Alvar Vinacua both for their participation in the initial ideas that generated this work, and for the very fruitful discussions and contributions during the whole preparation of the paper.

8. REFERENCES

[ABJN85] D.Ayala, P.Brunet, R.Juan, I.Navazo, "Object representation by means of non minimal division quadtrees and octrees", ACM Transactions on Graphics, vol 4, 1985, pp. 41-59.

[Aya88] D.Ayala, "Boolean operations between solids and surfaces by octrees: models and algorithm", Computer-aided design, vol 20, 1988, pp. 452-465.

[BFK84] W.Boehm, G.Farin, J.Kahmann, "A survey of curve and surface methods in CAGD", Computer aided geometric design vol 1, 1984, pp. 1-60.

[BrA87] P.Brunet, D.Ayala, "Extended octree representation of free form surfaces", Computer Aided Geometric Design, vol 4, 1987, pp. 141-154.

[BrN85] P.Brunet, I.Navazo, "Geometric modeling using exact octree representation of polyhedral objects", Proc. of Eurographics'85, North Holland, Nice 1985, pp. 159-169.

[BrN90] P.Brunet, I.Navazo, "Solid representation and operation using extended octrees", ACM Transactions on Graphics, vol 9., april 1990.

[CCV85] I.Carlbom, I.Chakravarty, D.A.Vandersschel, "A hierarchical data structure for representing the spatial decomposition of 3D objects", IEEE Computer Graphics and Applications, vol 5, 1985, pp.24-31.

[Dah89] W.Dahmen, "Smooth piecewise quadric surfaces", in Mathematical methods in Computer Aided Geometric Design, T.Lyche and L.Schumaker Eds, Ac. Press 1989, pp. 181-193.

[DuK89] M.J. Durst and T.L. Kunii, "Integrated Polytrees: A Generalized model for integrating spatial decomposition and boundary representation", in Theory and Practice of Solid Modelling, W.Strasser Ed., Proc. of Theory and Pract. of Solid Modelling, Tubingen, Springer Verlag 1989.

[Gar82] I.Gargantini, "Linear octrees for fast processing of three dimensional objects", Computer Graphics and Image Processing, vol 20, 1982, pp 365-374.

[Hof89] C.M. Hoffmann, "The Problems of Accuracy and Robustness in Geometric Computation", IEEE Computer Graphics and Applications, vol 22 num 3, 1989, pp. 31-41.

[KaE80] E.Kawaguchi, T.Endo, "On a method of binary picture representation and its application to data compression", IEEE PAMI vol 2, 1980, p. 27-35.

[Mea82] D.Meagher, "Efficient systhetic image generation of arbitrary 3D objects", Porc of the IEEE Computer society, Conf. on Pattern Rec. and Image Processing, 1982, pp. 473-478.

[Nav86] I.Navazo, "Geometric modeling of octree encoded polyhedral objects", doctoral thesis, Univ. Politecnica de Catalunya, Barcelona 1986.

[PlV90] N.Pla, A.Vinacua, "Algorithms for the generation of the canonic surface of a face octree", in preparation.

[Req80] A.Requicha, "Representations for rigid solids: theory, methods and systems", Computing surveys of the ACM, vol 12, 1980, pp. 437-464.

[Sam89] H. Samet, "The Design and Analysis of Spatial Data Structures", Addison Wesley Publ., 1989.

[SaW88] H.Samet, R.E.Webber, "Hierarchical data structures and algorithms for computer graphics; Part I: Fundamentals", IEEE Computer Graphics and Applications, vol 8, 1988, pp. 48-68.

[Sed85] T.W.Sederberg, "Piecewise algebraic surface patches", Computer aided geometric design vol 2, 1985, pp. 53-59.

CAPTIONS OF THE FIGURES

Figure 1. Face, Edge and Vertex nodes in the extended octree representation.

Figure 2. Face, Face-and-edge, and extended octree representation of a simple object.

Figure 3. The surface in a face node of a Face octree must be closer than ε to the corresponding associated plane.

Figure 4. The overall surface of the object being represented is contained in the union of a set of bands in the cubes associated to face nodes (2D example)

Figure 5. Discrete representation of normal vectors. In a), the set of normal vectors in the first octant. In b), the mesh of representable points in a triangle. Each of the coordinate directions belong to a single octant, c).

Figure 6. Octree representations of a solid object.

Figure 7. Example of surface map of the function d_face(P) in the case of a simple polyhedron.

Figure 8. Face octree representation of an object

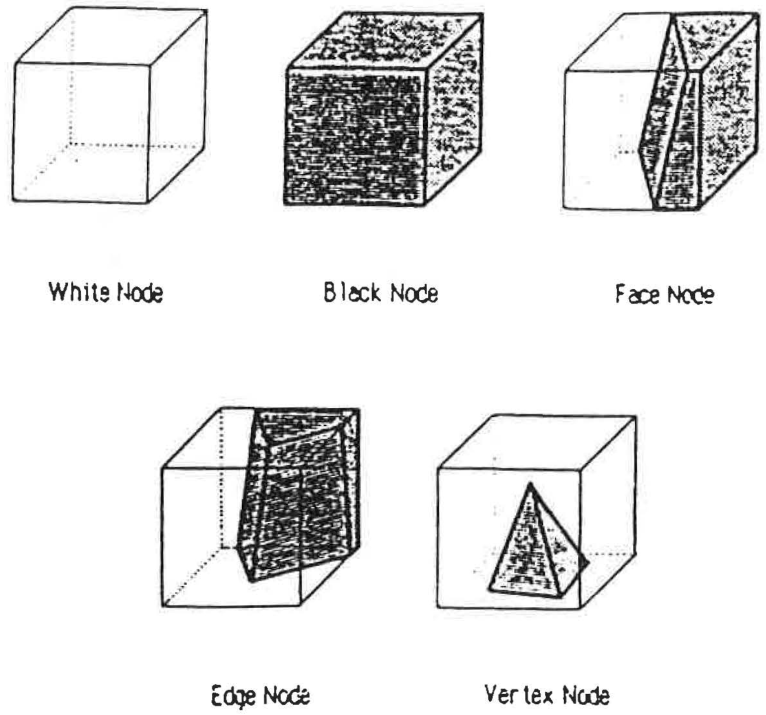

Figure 1. Face, Edge and Vertex nodes in the extended octree representation.

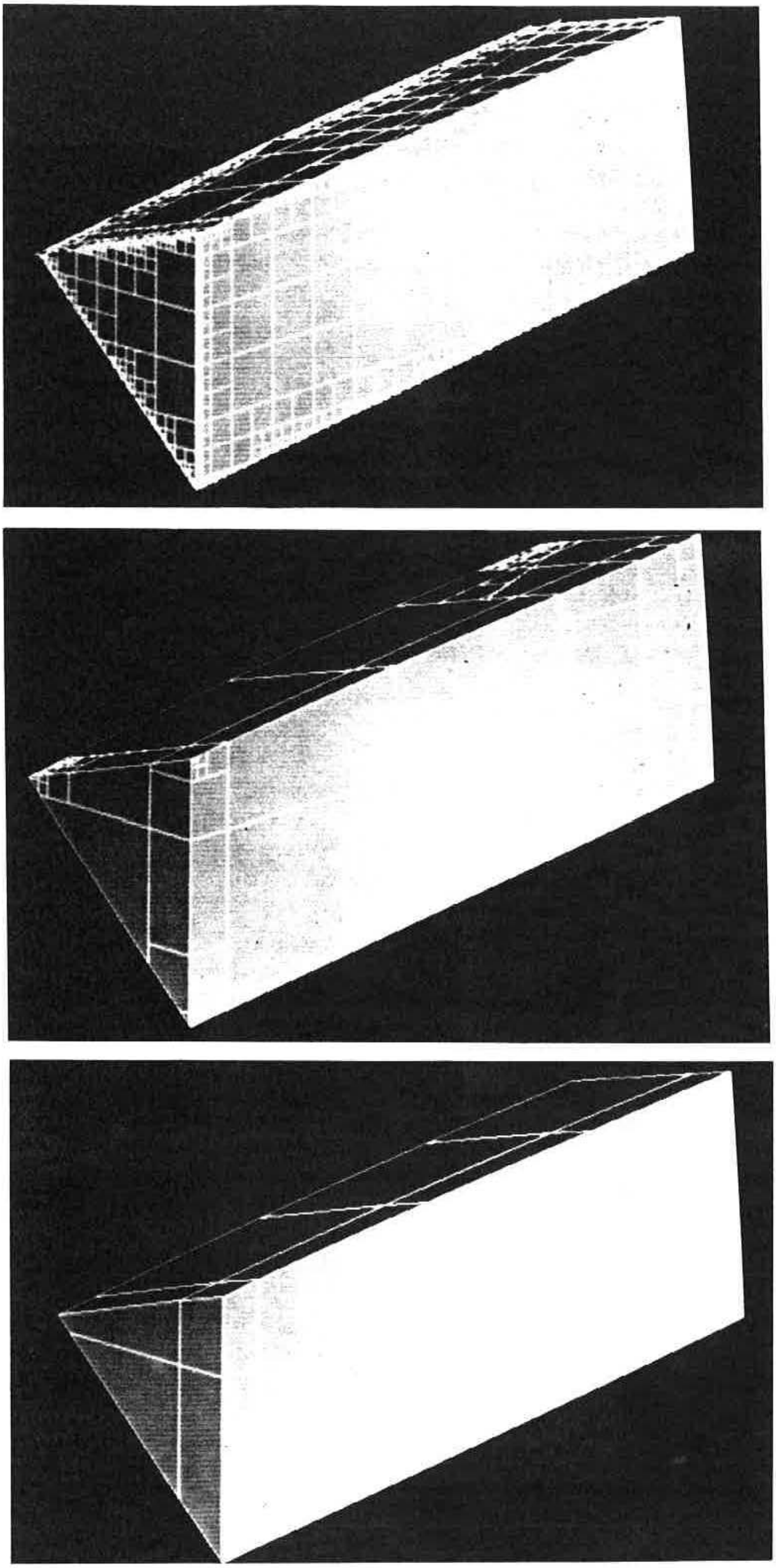

Figure 2. Face, Face-and-edge, and extended octree representation of a simple object.

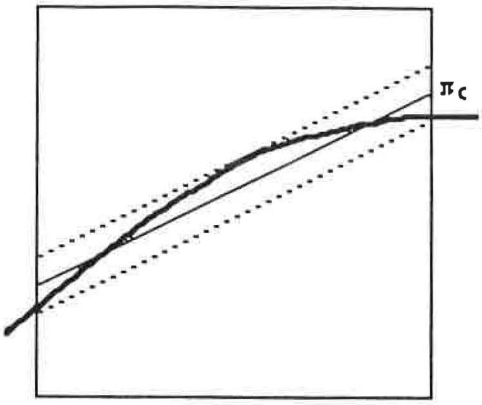

Figure 3. The surface in a face node of a face octree must be closer than ϵ to the corresponding associated plane π_c

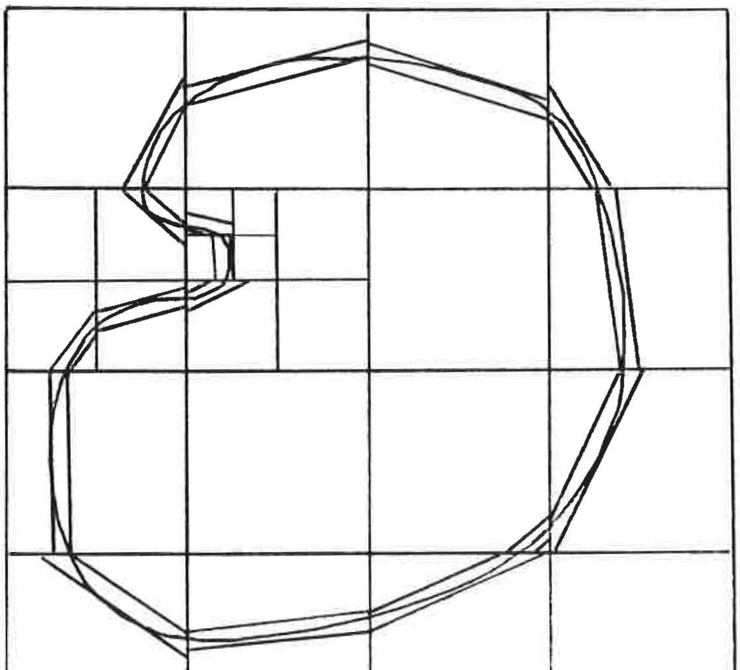

Figure 4. The overall surface of the object being represented is contained in the union of a set of bands in the cubes associated to face nodes (2D example)

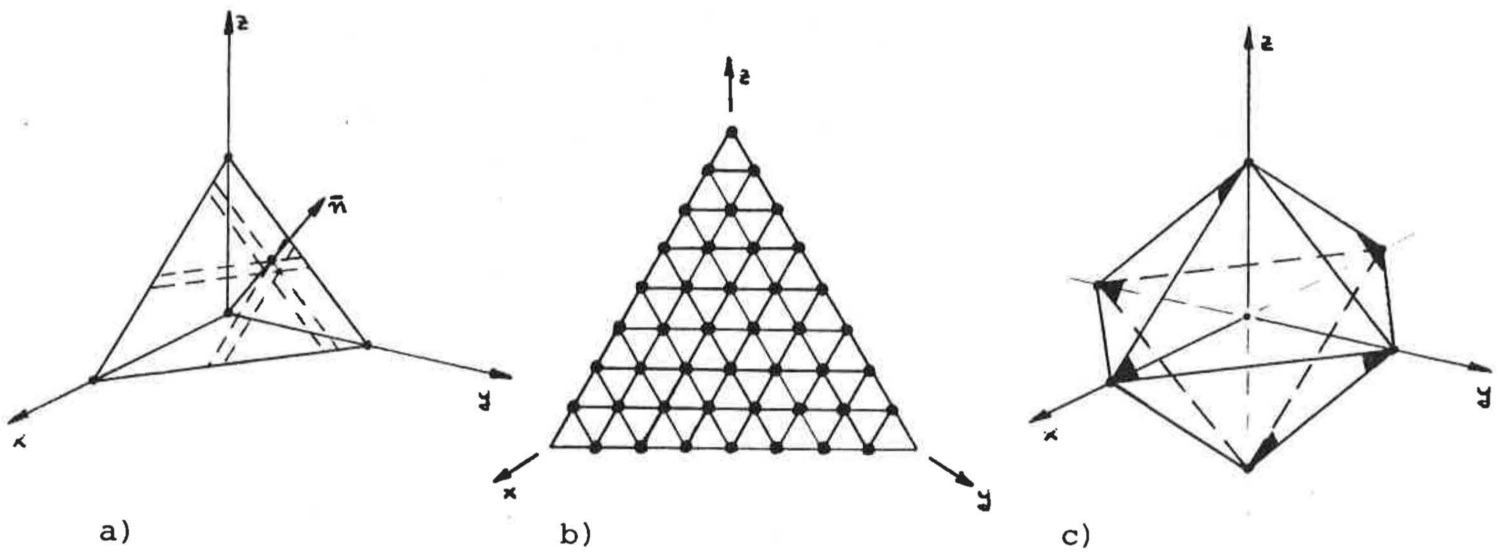

Figure 5. Discrete representation of normal vectors. In a), the set of normal vectors in the first octant. In b), the mesh of representable points in a triangle. Each of the coordinate directions belong to a single octant, c).

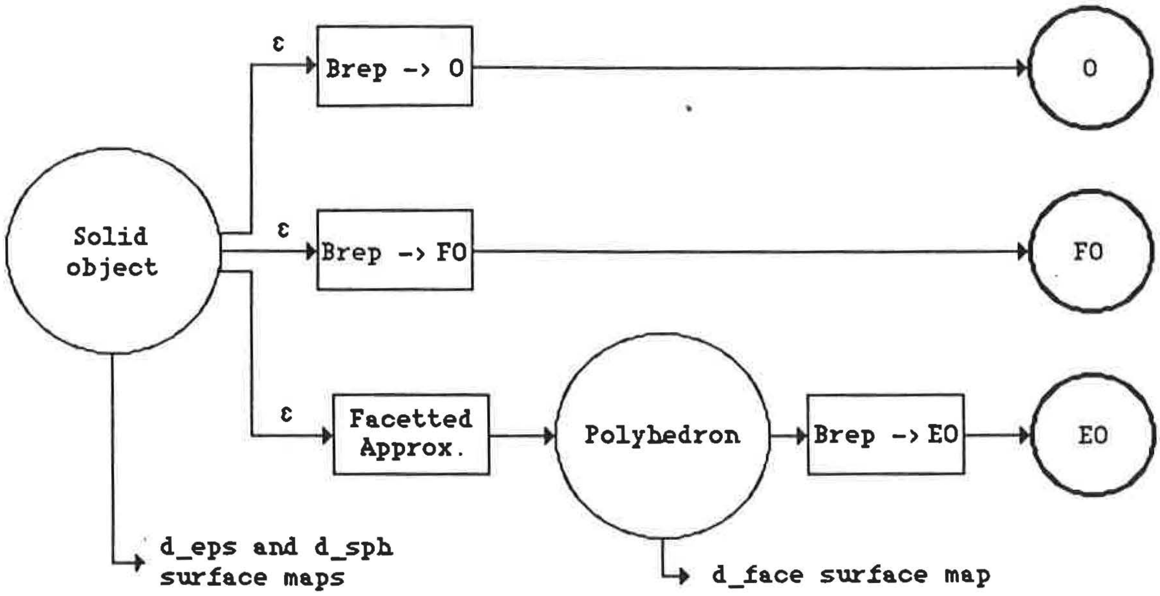

Figure 6 Octree representations of a solid object.

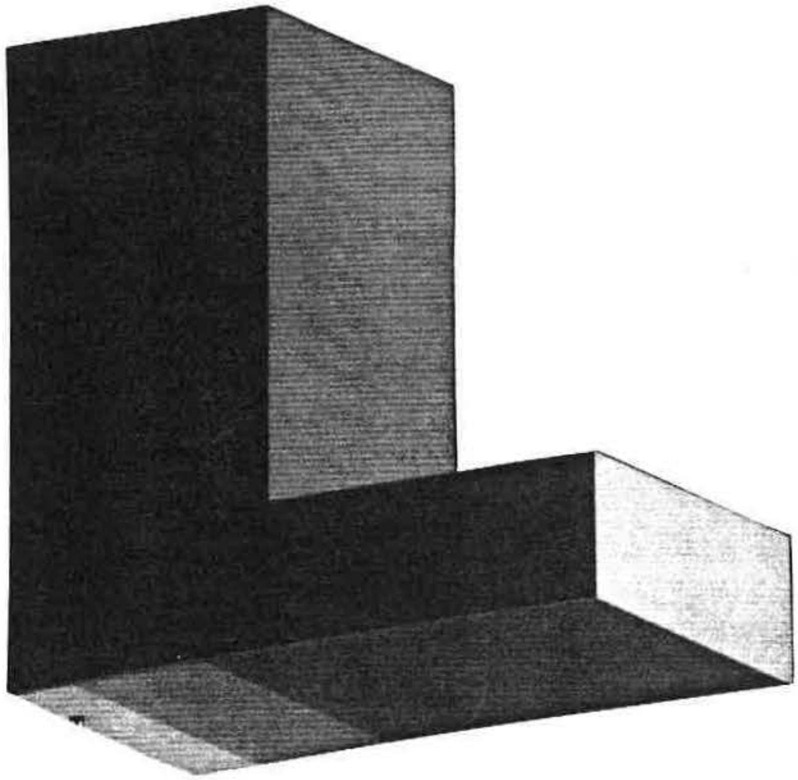

Figure 7. Example of surface map of the function d_face(P) in the case of a simple polyhedron.