TOPOLOGICAL ANALYSIS OF POLYHEDRAL OBJECTS

Zeinab E. Eid, Hussien H. Aly, and Mohammad S. Selim
Computer Science & Automatic Control Department, Faculty of Engineering,
Alexandria University, Alexandria, EGYPT.

ABSTRACT

In this paper, the domain of polyhedral objects is considered. A special kind of structural recognition that can be used to identify the solid object is presented. The technique is based on analyzing the topological characteristics of the solid object and decomposing this topology into primitive topologies. The inter-relationship between these primitive topologies are constructed as a tree structure. The analysis algorithm proposed, uses the boundary representation model of solid polyhedra and, it is shown to have a polynomial time complexity. It is shown to be sound and complete over a non-trivial subset of polyhedra that are characterized by having planar graph embeddings. The algorithm can be used to impose standardization of the process of solid objects encoding schemes and as a "hashing" mechanism in CAD/CAM systems and in information systems that contain a large database of polyhedra.

1. INTRODUCTION

The general problem of analyzing the shape of a solid object and addressing its related record in an information system is an important issue. For example, in CAD/CAM systems, this issue is essential for defining an automatic link between the design and manufacturing phases. Also, in the field of three-dimensional (3-D) object recognition, the existence of a well defined address space is very important for practical implementation of recognition systems [1], so that retrieval of objects out of large libraries can be efficient.

Systems that deal with solid objects, such as integrated software systems for CAD/CAM applications, usually deal with graphic models of these objects [2,3]. By analyzing these models; features, properties, materials, drawings; process planning and other information items can be derived or retrieved. Most of these information are related to the shape of the solid object. However, no automatic procedures exist that can generate these relationships. The way systems deal with this problem now, is by assigning artificial identification numbers to be used as a primary key to relate different information items about the same solid object. The generation of these identification numbers is done either as serial ones in the system, or by encoding the object according to some scheme. These are usually add-hoc and table look-up schemes that require experienced and qualified persons to get correct encoding [4,5,6].

In this paper, we are interested in the analysis process of polyhedral solid objects, i.e. solid objects whose all faces are planar ones. We propose a method to decompose an object into primitive objects with structural relationships between these primitives in the form of a tree structure. This decomposition is based on the connectivity properties of the edge-vertex graph model representing the topology of the solid object, and on some inference rules that are used to decompose this topology into primitive topologies.

In the next section, we present a quick survey of different techniques and graphic models that are used in analyzing solid objects. In section 3, we give the necessary terminology and mathematical background needed in the analysis process. Section 4 presents theorems that are required to establish the necessary and sufficient conditions that guarantee the correctness of the proposed analysis algorithm. The algorithm is presented in section 5 as a Pascal-like pseudo code. Finally, we give our conclusions in section 6.

2. SURVEY

A "feature" of a part is defined to be a recognizable geometric shape entity in the part, such as holes, slots, pockets, ...etc.Recently,many research papers appeared that concentrate on recognizing and extracting these features out of different graphic models, e.g., [7,8,9,10].

In [7], an interactive system to recognize shape features in 3-D solid models is described. Geometric, as well as, topological information are used in this process which is interactively guided by the user.

In [8], a feature extraction algorithm, based on topological information, is presented. Face-Adjacency-Graph (FAG) model is used to check for bi-connected and tri-connected components, that correspond to specific features in the solid object.

Peklenik at al., [9], have defined a Part Spectrum DataBase (PSDB) that contains the engineering models of different parts. The grouping of parts in the PSDB is based on their geometrical and technological aspects. Parts are classified and coded manually using special matrices of predefined features.

In the area of computer vision systems, [10] presented a system for recognizing 3-D objects from image views with range data. An Attributed Hypergraph Representation (AHR) is constructed and compared with another complete AHR of a prototype object.

Another trend for analyzing solid objects is by using the hyperpatch model [11], or the cell decomposition model [12]. These solid models are used extensively in 3-D finite element methods for the numerical solution of differential equations, and in computing various integral properties of solids such as volume and moments of inertia [12]. An example of this approach is presented in [13], where a decomposition method of a polyhedral solid object based on the triangulation of its boundary graph model is developed. This analysis method decomposes a solid into a set of disjoint tetrahedra. Although it may be efficient to use a single well defined primitive, such as the tetrahedron, in calculating some aspects related to the geometry or the mass of the solid, this will be of no help in designing the address space for these solids. We need a suitable large number of different primitives so that the address space is large enough to accommodate the different shapes encountered.

Feature extraction methods can be considered a special case of the solid object analysis approach, since only 'features' are recognized and extracted. So, we select the approach of decomposing solids into primitives as the basis of our research. An extensive survey of solid modeling and analysis methods can be found in [14].

3. TERMINOLOGY AND MATHEMATICAL BACKGROUND

An undirected graph G is defined as the couple (V,E), where V is a finite set of vertices, and E is a nonempty set of edges of the graph, $E \subseteq V \times V = \{\{x,y\} \mid x \in V, y \in V\}$. A vertex whose degree is three, (i.e., that have exactly three edges of E incident to that vertex), will be called a primitive vertex. A solid object whose vertices are all primitive ones is called a trihedral object. An n-sided pyramid is an object that has (n+1) vertices such that one of the vertices, called the apex of the pyramid, has degree equals n, while all other n vertices are primitive ones.

The graph G is called embeddable on a surface if it is possible to draw the graph on that surface such that no two edges of it intersect except at a vertex. Once the graph has been embedded, we can define faces as circuits of the graph that do not contain interior edges. In [15], it is proved that every planar graph has a unique embedding, (i.e., has a unique set of faces), on an orientable surface if, and only if, it is 3-connected. The famous Euler formula : V - E + F = 2 relates the number of vertices V, edges E, and faces F of a simple polyhedron. Two faces f1 and f2 are topologically equivalent if and only if they have the same number of edges and vertices.

Two or more polyhedral objects can be glued together to obtain another more complex polyhedron. In this paper, we consider only gluing of polyhedral solids through complete and topologically equivalent faces, such that the resulting polyhedron will have faces defined by a single loop of edges. In other words, gluing two objects S1 and S2 through faces f1 in S1 and f2 in S2, will result in a new object S where both faces f1 and f2 disappear, and the degrees of common vertices will change accordingly. Figure illustrates the gluing of a 4-sided pyramid with a cube

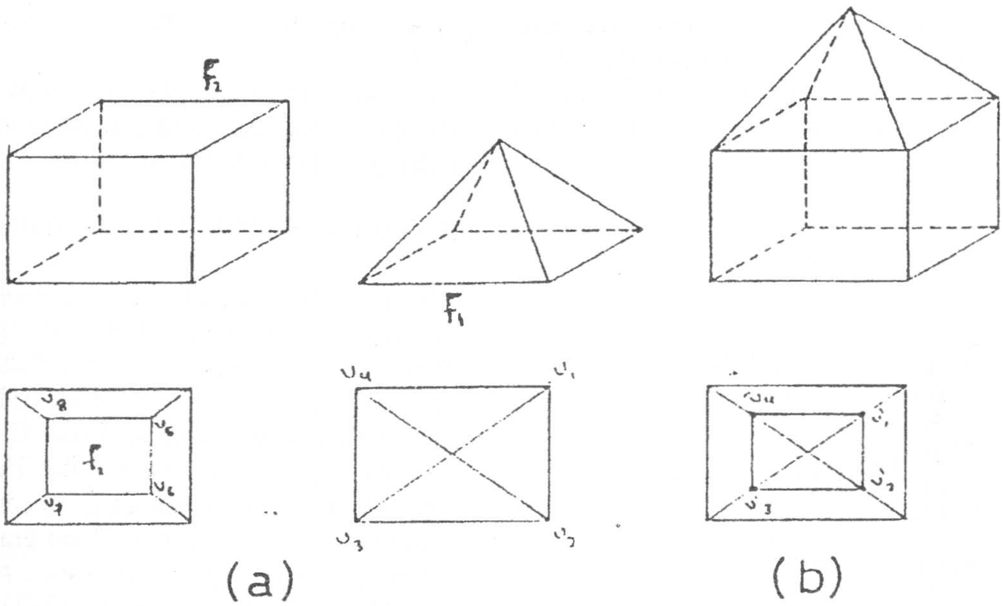

Figure 1. The glue of simple objects:
(a)- Two simple objects and their embedded graphs. (b)- Composite object.

4. DECOMPOSITION OF EMBEDDED GRAPHS

Given an embedded graph G of a composite object, we can scan the graph to search for a cycle of edges and non-primitive vertices in G, such that it can be considered as if it is a "cutting plane" through the solid. In other words, it can represent the gluing faces of two polyhedra. Splitting the edges and vertices of such a cycle, will divide G into two disconnected subgraphs G1 and G2. If both subgraphs G1 and G2 represent correct embedded graphs of polyhedra, (i.e., each subgraph is a tri-connected one), the original cycle will be called a mesh. Figure (2) is an example of a mesh.

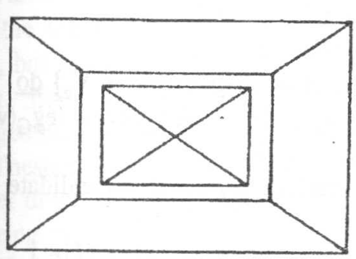

Figure 2. Splitting of a graph into two components through a mesh.

Obviously, not every cycle of edges and non-primitive vertices, is a mesh, and we need to characterize formally these meshes. The following theorem provides us with a necessary and sufficient condition for a cycle to represent a mesh. The proof of the theorem can be found in [14].

THEOREM 1:

A necessary and sufficient condition for a cycle in an embedded graph G (V,E) to represent a mesh is that no two edges of the cycle belong to the same face in the graph G.

Two meshes are said to overlap if they contain one or more common faces. An example of two overlapped meshes is shown in Figure (3), where the mesh m1 represents the embedded subgraph of a 6-sided pyramid, and the mesh m2 represents the embedded subgraph of a 4-sided pyramid. Meshes m1 and m2 overlap at two faces f1 and f2.

The overlapping of meshes presents a serious problem for the decomposition approach, since it leads to a non-deterministic output. For example, in Figure (3), we can get either the 6- sided or the 4-sided pyramid as an output. To avoid this non- determinism, and to ensure the uniqueness of the output, we have decided to consider the union of all overlapped meshes. The following theorem states that the union of overlapped meshes is also a mesh. for proof, refer to [14].

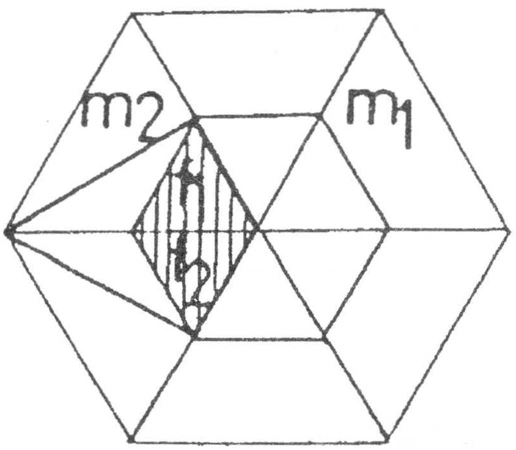

Figure 3. Two overlapped pyramids.

THEOREM 2

The union of two or more overlapped meshes will result in a new cycle which is a mesh.

The decision to consider the union of overlapped meshes, means that the output will contain only disjoint meshes. Hence, the analysis algorithm will allow non-trihedral objects to be recognized as primitives. This is good, since it makes the set of primitives an open set, and will in turn, widen the address space allowed. In this way, we ensure to get the same output, and hence the same key-value, each time the same polyhedral object is to be analyzed. Non-trihedral objects that will be recognized as primitives are of no harm, since we use these primitives only in encoding the original object.

Pyramids are considered to be primitive objects, since they have well defined structures. The following theorem helps in recognizing the graphs of overlapped pyramids.

THEOREM 3 :

Two subgraphs that can be recognized as embeddings of two pyramids overlap at exactly two triang[...] faces.

Three of such subgraphs overlap at exactly a si[...] triangular face. No more than three of such subgr[...] overlap in a common face.

5. DECOMPOSITION ALGORITHM

In this section, we present the proposed algorithm [...] decomposition of polyhedral solid objects. Figure [...] shows the procedure DECOMPOSE which i[...] recursive representation of the algorithm. The inp[...] this procedure is a data structure G representing [...] embedded graph of the input solid. The output is a [...] structure T where each node is an embedded grap[...] a solid. Leaves of the tree T are graphs of primi[...] solids. For optimization purposes, a reduced grap[...] is obtained from the original graph G by eliminating [...] primitive vertices and their incident edges. [...] procedure recursively processes the input graph [...] extracting the possible primitives at each level.

Procedure Decompose (G,T);
<u>begin</u>
T:=G;
f_o:= Embedding_face(G);
$V' := \{v \mid v \in V$ <u>and</u> $\deg(v) > 3\}$;
$E' := \{(v_i,v_j) \mid (v_i,v_j) \in E$ <u>and</u> $v_i,v_j \in V'\}$;
G':=Construct_a_graph(V',E',F');
<u>if</u> $E'=\phi$ <u>or</u> $F'=\phi$ <u>then</u> exit;
bi_connected_components(G',components);
primitives:=ϕ;
<u>for</u> each $G_c(V_c,E_c,F_c) \in$ components <u>do</u>
<u>begin</u>
f_o':=Embedding_face(G_c);
$F_c := F_c - \{f_o'\}$;
meshes:=$\{m_i \mid m_i \in F_c$ <u>and</u> $m_i \notin F\}$;
Valid_meshes:=$\{m_i \mid m_i \in$ meshes $\mid m_i \cap f_j \mid \leq 1 \ \forall\, f_j \in F\}$;
candidate_pyramid:=ϕ;
<u>for</u> k=4 to max $\{\deg(v) \mid v \in V_c\}$ <u>do</u>
<u>for</u> each $v_i \in \{v \mid \deg_{Gc}(v)=\deg_G(v)=k\}$ [...] test-pyramid(v_i,G_c,b_i)
<u>then</u> candidate_pyramid:=candidate_pyrami[...] $\{(v_i,b_i)\}$;
pyramids:=test_overlap($\{b_i \mid (v_i,b_i)$ [...] candidate_pyramid$\}$);
valid_meshes:=valid_meshes U pyramids;

```
V_c':=V_c - {v | (v,b) ∈ candidate_pyramid};
E_c':={(v_i,v_j) | (v_i,v_j) ∈ E_c and v_i,v_j ∈ V_c'};
G_c':=construct_a_graph(V_c',E_c',F_c);
f_o':=embedding_face(G_c');
invalid_meshes:=meshes - valid_meshes;
corrected_meshes:=φ;
for each m_i ∈ invalid_meshes do
  if correct_mesh(m_i,new_mesh,F,valid_meshes)
  then corrected_meshes:=corrected_meshes U new_mesh;
  valid_meshes:=valid_meshes U test_overlap (corrected_meshes);
  if valid_meshes= φ
  then
    if f_o'≠f_o then
      if ( | f_o'∩f_j | ≤1, ∀f_j ∈ F)
      then valid_meshes:= f_o'
      else if correct_mesh(f_o',new_mesh,F,φ)
      then valid_meshes:=new_mesh;
    else exit;
primitives:=primitives U valid_meshes;
end; (*for each G_c *)
if primitives=φ then exit;
for each m_i ∈ primitives do
begin
  p_i:=build_a_primitive_graph (m_i,G);
  T_i:=p_i;
  add_subtree(T_i,T);
end;
adj_matrix(T_i,M_i);
G':=G;
decompose(G',T');
add_subtree(T',T);
end; (*decompose procedure *)
```

Figure 4. Main procedure of analysis algorithm.

We should notice that, since we are dealing only with topological information, i.e., no metrics (geometry) is present in the input data, objects that have different geometries, but same topology will be considered the same thing. For example, Figure (5) shows three different geometries corresponding to the same topology. These three objects will be considered as a single object in our analysis approach.

By analyzing the procedure, we can show that it is $O(|E|^3)$ [14]. Figure (6) gives an example of an input graph embedding of a polyhedral object (root) and the resultant primitives (leaves). The labels on the tree branches represent structural relationships between primitives and their parent objects.

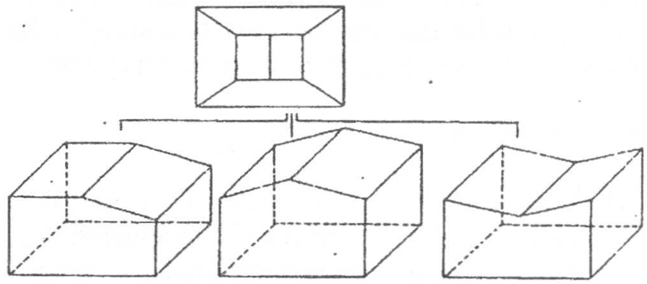

Figure 5. Single topology with different geometries.

information system.

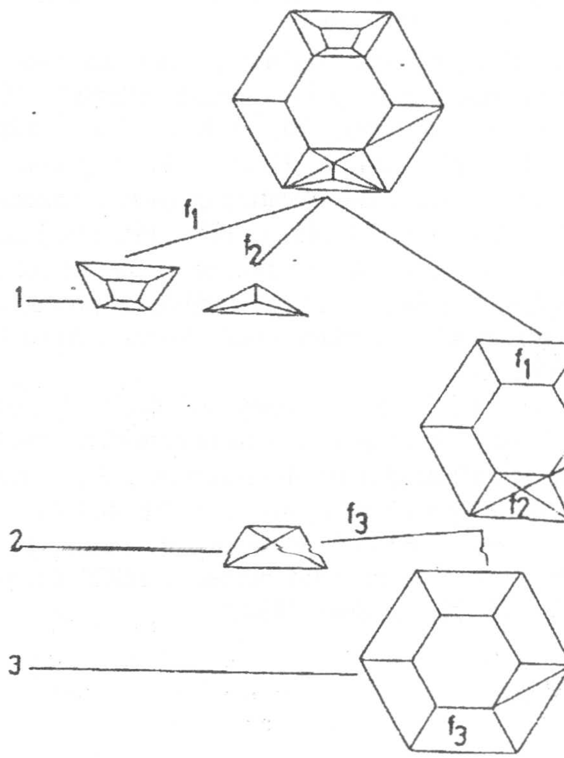

Figure 6. An example

6. CONCLUSION

We present an algorithm of $O(|E|^3)$ that decomposes polyhedral solids into primitive ones, based only on topological information. We provide theorems that ensure the correctness of the algorithm over the set of polyhedra that is characterized by having planar graph embeddings. The output of the analysis algorithm is a tree structure that can be considered as a "finger print"of the input solid. This tree can be used to derive an identification figure for the input solid that can be used to retrieve related records in the database of an information system.

REFERENCES

[1] W.C. Lin, F.Y. Liao, C.K. Tsao and T. Lingutla, "A hierarchical multiple-view approach to three-dimensional object recognition", *IEEE Trans. Neural Net.*, vol. 2, No. 1, Jan. 1991.

[2] S. Dulli and V. Milanese, "A database design for graphical models", *ACM SIGPLAN Notices*, Vol. 25, No. 12, Dec. 1990.

[3] A. Kemper and Wallrath, "An analysis of geometric modeling in database systems", *ACM Computing Surveys*, Vol. 19, No. 1, March 1987.

[4] J. Peklenik and j. Grum, "Investigation of computer aided classification of parts", *Annals of the CIRP*, Vol. 29, No. 1, 1980, PP. 319-323.

[5] T.A. Sparrow, "Application of group technology and casting classification in small-batch-production foundaries", *Foundary Trade Journal*, April 13, 1972.

[6] G.H. Middle, R. Connolly and R.H. Thornley, "Some design and development considerations for an industrial classification system", *Proc. Instn Mech Engrs* 1970-71, Vol. 185, PP. 607-619.

[7] H. Sakurai and D.C. Gossard, "Recognizing shape-features in solid models", *IEEE CG&A*, Vol. 10, No. 5, Sept. 1990.

[8] L. De Floriani, "Feature extraction fr boundary models of three dimensional objec *IEEE Trans. Pattern Analysis & Mach. Int* Vol. 11, No. 8, Aug. 1989, PP. 785-798.

[9] J. Peklenik and R. Sekolonik, "Developmen the part spectrum database for comp integrated manufacturing systems (CIMS *Annals of the CRIP*, Vol. 39, No. 1, 1990.

[10] A.K.C. Wong, S.W. Lu and M. Rio "Recognition and shape synthesis of 3-D obj based on attributed hypergraphs",*IEEE Tr Pattern Analysis & Mach. Intell*, Vol. 11, 3, March 1989.

[11] M.S. Casale and E.L. Stanton, "An overview analytic solid modeling", *IEEE CG&A*, Vol No. 2, Feb.1985.

[12] A. Requicha, "Representations for rigid sol Theory, Methods, and Systems", A *Computing Surveys*, Vol. 12, No. 4, Dec. 19 PP. 437-464.

[13] T.C. Woo, "Interfacing solid modeling to C and CAM: Data structures and Algorithms decomposing a solid", *IEEE Computer*, Vol. No. 12, PP. 44-49, Dec. 1984.

[14] Z.E. Eid, "Graph-based analysis of polyhe objects", *Ph.D. Dissertation*, Computer Sci & Auto. Control Dept., Faculty of Eng., A Univ., EGYPT, Feb. 1991.

[15] P.R. Hanrahan, "Creating volume models f edge-vertex graphs", *ACM SIGGRAPH*, Vol. No. 3, Jul. 1982.