Journal of Universal Computer Science, vol. 9, no. 9 (2003), 1008-1029
submitted: 10/2/03, accepted: 9/6/03, appeared: 28/9/03

Abstract Representation of Object and Structural Symmetries Detection

Vincent Dugat
(IRIT-UPS, Toulouse, France
dugat@cict.fr)

Pierre Gambarotto
(IRIT-UPS, Toulouse, France
gamba@irit.fr)

Yannick Larvor
(IRIT-UPS, Toulouse, France
larvor@irit.fr)

Abstract: This paper describes a method for constructing an abstract representation of a shape from a classical polyhedral 3D representation of an object. This framework is suitable for qualitative reasoning. As an application we use this abstract representation to compute the structural symmetries of a 3D polyhedron. The starting point of the computation is a classical polyhedral 3D representation of the object. From the Medial Axis Transform (MAT) of this object we propose a more abstract representation based on a set of spheres extracted from the MAT and structured as one or several graphs. This framework can be used for several purposes. Here we focus on the problem of finding structural symmetries of the object. We use the automorphisms group of the computed graphs. Then we propose a method to compute the automorphisms that have a geometrical sense among the set of all automorphisms. We compare the brute force algorithm with a branch and bound strategy based on the orbits partition of the vertices.

Key Words: Qualitative reasoning, spatial reasoning, shape recognition, spatial representation, medial axis transform.
Category: I.2.4, I.3.5.

1 Introduction

Two majors fields in AI are concerned with representation of objects: robotics on the one hand, and qualitative spatial reasoning (henceforth QSR) on the other. Robotics' analysis tools are traditionally issued from classical physical and mathematical science. Such tools are especially accurate to process raw numerical data coming from robot's sensors. On the contrary, QSR focuses on more abstract representations, and puts the emphasis on qualitative relations and expressive power rather than numerical precision. This manichean view is not satisfying, for many reasons: robots sensors will (at least for some time) continue to provide numerical data, but needs in intelligent behavior require higher levels of abstraction, for instance in order to interact with human language, or for sophisticated pathfinding algorithms.

There still remains a gap between the result of the treatment of numerical data (by various techniques such as mesh reconstruction, boundary extraction, etc.) and qualitative reasoning on abstract entities.

In this article, we propose a method to construct an abstract representation merging numerical and abstract informations. We start from objects represented in two or three dimensions: each object is built from a set of vertices, and each vertex is located with coordinates in a Cartesian space. The final objective is to express the structure of such an object in an abstract framework. As an application we expose a method to express the structural symmetries of a 3D convex polyhedron using the constructed framework. Note that this abstract framework allows different levels of granularities in this kind of computation. The starting point is a classical 3D representation of the object. We shall first expose the abstract representation of the shape: from the Medial Axis Transform (MAT) of this object is extracted a set of spheres structured in a graph which can be used for several purposes such as shape identification or shapes comparison. In this paper we shall focus on the problem of finding the symmetries in the structure of the shape. We shall used the graph representation and tools of algebraic graph theory such as a branch and bound algorithm based on automorphism group results. Sections 2 exposes the construction of the Medial Axis Transform and section 3 focuses on the construction of the structure graph and other associated graphs extracted from the MAT in order to construct our abstract representation. As an application, section 5 exposes a graph-theory based method for generating automorphisms that have a geometric sense opposing the brute force algorithm to an ad-hoc branch and bound one. Last, we precise that in the rest of this paper the terms form and shape will be synonymous.

2 Constructing the MAT

The first step of our representation is to construct the medial axis transform. We shall detail this construction in the subsections that follow.

2.1 MAT: definition

The computation of the MAT is a complex task. We first define some notations before detailing the construction algorithm:

- $\mathbb{R}$ represents the set of real numbers; $\mathbb{R}^3$ denotes the standard Euclidean space.
- The word "distance" denotes hereafter the Euclidean distance in $\mathbb{R}^3$.

- Let c be a point of $\mathbb{R}^3$ and $r \in \mathbb{R}$. The set of points whose distance from c is less than or equal to r is called the (closed) ball of center c and radius r.
- In this paper, only the 3-dimensional case is considered. The 2D case can be treated with only minimal changes in the vocabulary used (mainly replace "ball" with "disc").

The *Medial Axis* (MA) of an object is defined as the locus of the centers of its maximal interior balls. A ball inside an object is maximal iff it is contained in the object and is not a subset of another ball inside the object[1]. The *radius function* denotes the function that associates to a point of the MA the radius of the corresponding maximal ball. The *Medial Axis Transform* of an object is the MA of this object together with the associated radius function.

With the notations introduced above, this definition becomes:

Definition 1 Let S be a subset of $\mathbb{R}^3$. The *Medial Axis* of S, noted $M(S)$, is the set of the centers of all maximal ball inside S, together with the limit points of this set.

Definition 2 The value at each $M(S)$'s point of the *radius function* r is the radius of the associated maximal ball. The values of this function are in $\mathbb{R}$, and r is continuous on $M(S)$.

Definition 3 The *Medial Axis Transform* of the subset S of $\mathbb{R}^3$ is constituted by $M(S)$ and r.

2.2 Computation of MAT

In this paper, and for the sake of simplicity, we only consider the case of polyhedra with simply connected face[2]. This restriction can be bypassed by subdividing multiply connected faces into simply connected faces. Such a decomposition is described in the appendix B of [Sherbrooke *et al.*, 1996]. The algorithm used to compute the MAT of a polyhedra is a slightly simplified version of [Sherbrooke *et al.*, 1996]. This algorithm has the advantages of being exact and of having a polynomial complexity (in terms of number of faces).

2.2.1 Classification of MAT's points

Each particular point of the MA is linked to several faces, as explained in the following definition:

[1] Although we consider Euclidean 3-dimensional space, the only prerequisite of this definition is a metric space

[2] Intuitively, a face is simply connected if it has no hole.

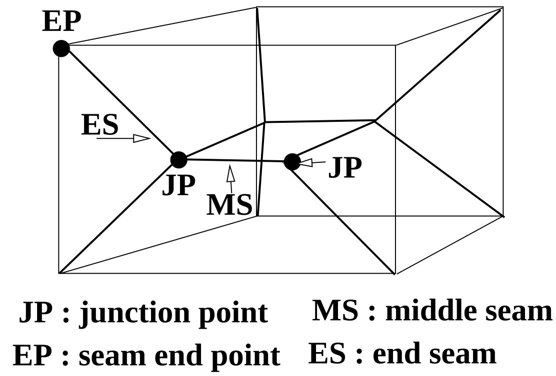


Figure 1: MA Points

Definition 4 Let $\mathbf{p}$ be a point on the MA of the polyhedral P. Let $B_r(\mathbf{p})$ be the corresponding maximal sphere centered on $\mathbf{p}$, of radius r. Let $\mathbf{q}_1,\ldots,\mathbf{q}_n$ be the points on the boundary of the object where $B_r(\mathbf{p})$ is tangent. The $\mathbf{q}_i$ are called footpoints of $\mathbf{p}$. If j_i designates the index of the face, edge or vertex on which $\mathbf{q}_i$ lies, then P_{j_i} is said to be a *governor* of $\mathbf{p}$.

When considering only non degenerated cases, a particular point of the MA is obtained for each set of governors. With the notations introduced above, the specific points of the MAT used in our algorithm[3] are:

Junction Point : defined by four non degenerated governors.

Seam Point : defined by three non degenerated governors. For given governors, the set of seam points is generally a curve called a *seam*.

Seam-End Point : intersection of the boundary of the object and a seam of the MAT.

Sheet Point : defined by two non degenerated governors. Connected sheet points form a surface, called a *sheet*.

End Point : intersection of the boundary of the object and a sheet. A connected set of end points is called a *rim*.

End seam : a seam between a junction point and a seam-end point (i.e. a vertex of the polyhedron).

In figure 1 are represented particular points of the MA of a parallelepiped.

[3] For more information on the MAT algorithm and further details in the classification of points of the MAT, refer to ([Sherbrooke, 1995]).

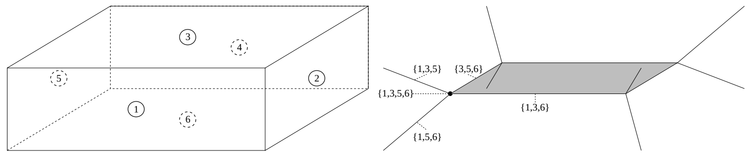


Figure 2: MAT of a rectangular box

2.2.2 Principle of Sherbrooke's algorithm

The above classification is primordial for the algorithm computing the medial axis as it is shown in [Brand, 1992], the behavior of the MA near a MA point is completely determined by the configuration of the point's footpoints, or in other terms by the set of governors of this point. For instance, a junction point always lies at the junction of seams and a seam is at the intersection of sheets. This relation is made more evident by considering the governors set of each of these different elements. For instance, a seam is obtained with a set of three governors in the general case. Two intersected seam will have two governors in common, and the junction point will thus have four governors. More generally, the governors' set of an intersection is obtained by the union of the governors' set of the intersected parts of the MA.

The idea of the algorithm developed in [Sherbrooke *et al.*, 1996] is to use the reverse property: when you have found a junction point, you calculate its set of governors (by looking at its footpoints). Each subset of the junction point's set of governors is then checked to determine if it governs one of the seams intersecting at the junction point.

For the check, an initial tangent (i.e. at the junction point) is found by derivation of the system of governors' equations. The building of the seam is then achieved by integration of this system.

This algorithm is particularly simple when considering a convex polyhedra. In this case, all the parts of the MA connecting particular points are straight lines (and not general curves), and the corresponding system of equation is trivial. The integration can even be symbolically resolved.

At the vicinity of a concavity, the result of this MAT algorithm is generally much more complicated to obtain, because the MA may be locally a non degenerated curve or a non degenerated surface (i.e. not a line or a plane). The calculus of those elements can be obtained by an approximated method of integration (for instance by a Runge Kutta method).

Example: on figure 2 is represented a rectangular box and the associated MA (without the limit points constituted by the rims and the seam end points). The highlighted junction point's set of governors is $\{1, 3, 5, 6\}$. To find the possible seams intersecting at this point, we check every subset of three elements: $\{1, 3, 5\}$,

$$\begin{cases} x= \ r & [F][dx\ dy\ dz\ dr]^T = [0\ 0\ 0]^T \ (2) \\ y{=}r\ (1) & with\ [F] = \begin{bmatrix}1\ 0\ 0\ -1\\0\ 1\ 0\ -1\\0\ 0\ 1\ -1\end{bmatrix} \ \ (3) \\ z= \ r & \end{cases}$$

Figure 3: Differential system

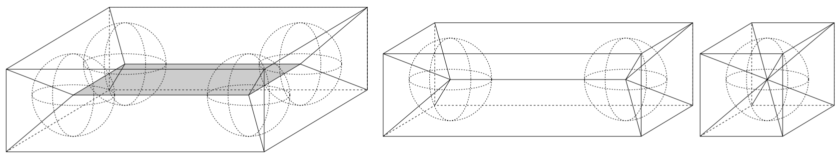


Figure 4: Degeneration cases

$\{1, 3, 6\}$, $\{1, 5, 6\}$ and $\{3, 5, 6\}$.

Let's consider the seam governed by the faces $\{1, 5, 6\}$. Let $(x, y, z, r)^T$ be a point of this seam (r is the radius of the associated maximal ball, or in other terms, the distance to the governors). A point of this seam must satisfy the system 2.2.2 (equation 1).

This system cannot be directly solved (3 equations, 4 variables). The solution is to find a tangent to the seam by integrating the corresponding differential system 2.2.2(equations 2 and 3)[4]. The result is for this example $(1, 1, 1, 1)^T$. We can then trace the seam, starting from the junction point, step by step. At each step, we verify if the seam is still correct.

This simple case is a non degenerated one, i.e. each subset corresponds to a real seam. As explained above, in the case of a convex polyhedra, the tangent found is constant (i.e. the corresponding seam is straight).

The figure 4 illustrates the possible cases of degeneration. On the left is the general non degenerated case (a sheet is determined by two governors exactly, and thus cannot be degenerated). In the center, the extremities of the central seam are degenerated (5 governors instead of the usual 4). The sheet previously found has become a seam. And finally, on the right of the figure, the most degenerated case is the cube: the 6 faces are governors of the cube, so the previously center seam is now reduced to a single point.

2.2.3 Treatment of concavity

The previous algorithm can compute the MAT near a concavity, but the result is expensive and not useful for our aimed tasks. So we simplify the treatment

[4] Method: use the Singular Value Decomposition of F. The desired tangent vector is associated with the singular value 0.

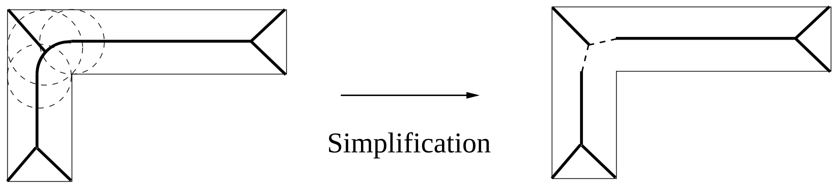


Figure 5: Simplification of concavity

of concavity in the MAT, by substituting the curved portion of the MA by the extreme points of that portion. The result is shown figure 5 in the 2D case. In the general case, the portion of the MA governed by a concavity is a quadratic surface. We substitute this surface by a set of points, following this algorithm:

1. Each intersection of the surface and a seam (junction point on the boundary of the surface) is kept back.
2. The extrema of the surface are computed, and the corresponding points are kept back.
3. The quadratic surface is deleted, replaced by the points defined by the two previous rules.

3 Using Mat for extracting information on the structure of a 3D objects

In fact, we shall not exploit directly the result of the MAT algorithm. In this section we present an algorithm to model a 3D object with a set of spheres extracted from the MAT. The method has some similarities with shock graphs [Macrini *et al.*, 2002] but the representation is different. The objectives of our representation are:

- to be as simple as possible.
- to be more adapted to study the structure of the object.
- to emphasis structural informations (precision is not the prime factor).
- to be incremental: if f^i measures the error at the stage i of the process of representation, then $f^{i+1} < f^i$. In other words, the algorithm behind the process has to be any-time.

Therefore the interpretation of the last aim becomes: in the computation of the MAT some spheres are more significative than others. The term "significative" is to be interpreted in the volumic sense, i.e. the most significative spheres are the biggest one. Such a sphere must be internally tangent to the object to at least 4 points in 3D. If not, it is trivial to find another sphere bigger (by slightly shifting the center of the sphere for instance, in order to make the radius grow).

Using the MAT we now construct a more abstract representation of a shape that is more suitable for symbolic manipulations. This needs some knowledge in graph theory so we shall begin by defining the vocabulary useful for this section.

3.1 Graph theory vocabulary

We suppose the reader familiar with graph theory and we just precise the vocabulary we use is the rest of the paper. Refer to graph theory books as [Berge, 1973] or [Godsil and Royle, 2001] for more informations on this topic.

A *labeled graph* is a sextuplet (V, E, L_v, L_e, LV, LE) where V is the set of vertices, $E \subset V \times V$ is the set of edges and $L_v : V \mapsto LV$, $L_e : V^2 \mapsto LE$ are two functions: L_v associates a label to each vertex and L_e associates a label to each edge of the graph. LV and LE are the sets where vertices and edges take their label values. These sets can be Cartesian products of other sets. Labeling functions define equivalence classes on vertices and edges. We can associate a color, i.e. an element of a color set C, to each class in order to obtain a colored graph:

A *colored graph* $\mathit{\Gamma} = (V, R_0, \ldots, R_d)$ is a set of vertices V and a set of relations of same arity $r, R_i, 0 \leq i \leq d$ with the properties:

1. $R_i \cap R_j = \emptyset$, relations are mutually disjoint,
2. $R_i \cap Diag(V^r) \neq \emptyset$ implies $R_i \subseteq Diag(V^r)$, where $Diag(V^r)$ is the diagonal of V^r,
3. $V^r = \bigcup_{i=0}^{d} R_i$.

Relations $R_i \subseteq Diag(V^r)$ represent vertex colors, while other relations represent edge colors. Non-edges receive a color such that a colored graph is always a complete graph. R_i is called the i-th color class of $\mathit{\Gamma}$.

$S(V)$ will denote the automorphism group of V, and $Aut(\mathit{\Gamma})$ will denote the subgroup of $S(V)$ preserving the relation of connection, i.e. $(a, b) \subset V^2$, $\sigma \in Aut(\mathit{\Gamma}) \Rightarrow E(a, b) \leftrightarrow E(\sigma(a), \sigma(b))$. O is an orbit of $\mathit{\Gamma}$ iff for two elements of O an automorphism can be found that transforms one of the element in the other: $\forall(x, y) \in O^2\ \exists\sigma \in Aut(\mathit{\Gamma})/\sigma(x) = y$. Let $\sigma \in Aut(\mathit{\Gamma})$, the order of σ is denoted by $o(\sigma)$ and is defined by: $o(\sigma) = n \leftrightarrow \sigma^n = Id$ and $\forall m$ such that $(0 < m < n)\ \sigma^m \neq Id$. Orbits collect vertices that have the same structural role in the graph and we have the following results:

Proposition 1 An orbit is invariant by an automorphism of $Aut(\Gamma)$: O is an orbit of $\Gamma \Rightarrow \forall \sigma \in Aut(\Omega) \;\; \sigma(O) = O$.

Proposition 2 An automorphism σ can be entirely described by its cycles. The order of σ is the *lcm* (least common multiple) of the lengths of the cycles of σ.

An automorphism of order 2 is said to be an *involution*. Let $\sigma \in S(V)$. The ordered set $c = \{a_0 \; a_1 \cdots \; a_{n-1} \; a_n\}$ is a σ cycle of length n iff $c \subset V$, $\forall i \in [1 \cdots n] \; \sigma(a_i) = a_{i-1}, \;\; a_{i-1} \neq a_i$ and $a_n = a_0$. Least, let $\sigma \in S(G)$. $O \subset V$ is said to be fixed with respect to σ iff: $\forall x \in O \;\; \sigma(x) = x$. This notion can be extended to a set of vertices called a *block*.

Computing the automorphism group of a graph is a hard task. Even if the problem is not known to be NP-complete there is at this date no polynomial algorithm. Fortunately there are efficient programs based on arborescent technics such as [McKay, 2000] or the Weisfeiler-Leman (WL) Algorithm [Babel *et al.*, 1997].

A *planar graph* is a graph that can be drawn without crossing edges. Hopcroft and Wong established in [Hopcroft and Wong, 1974] that computing the automorphisms group is polynomial for planar graphs.

3.2 Sphere representation: construction of the structure graph

Sherbrooke's algorithm for computing MAT works on any polyhedron as explained in section 2.2.3. As said at the end of the same section we introduce a simplification of the MAT near a concavity. This reduces the number of spheres needed to represent a concavity and allows an easier detection of maximal convex parts. In fact, we can observe that there is one or two spheres for each concavity of the object. We call theses spheres: *articulation spheres*.

The result of our representation is derived from a subset of the MAT, i.e. a set of $B_r(\mathbf{p})$ spheres (Cf. def 4). The spheres are selected in three consecutive stages. The second stage deals with end-seams and it could loop until the desired granularity is reached. In fact, in such a process, all the spheres of the end-seam will be tangent to the previous and next computed ones. Moreover, their centers will be on the same line and the ratio of two consecutive spheres is the same. So we do not need to loop in the second stage and compute only the two first spheres on the end-seam: the first one to compute the ratio of their sizes (Cf. figure 6).

The core set of spheres is given by the spheres centered on junction point. For a more accurate precision, more spheres are added on seams pointing to a vertex of the polyhedron:

The following algorithm is an extension of the one of [Dugat *et al.*, 2002] and is able to compute a representation of any three-dimensional shape.

Notation 1 Let P denotes a polyhedron, PJ the set of Junction Points of P, $convhull(S)$ the convex hull of the set of spheres S, let $\overline{(s_i, s_j)}$ (where $((s_i, s_j) \in PJ^2$ and $i \neq j$) designs the middle seam between the junction points s_i and s_j. Let $E(s_i)$ denotes an End-Seam where s_i is the junction point that determines the seam.

1. $Result = Result \cup \{s | s \in PJ\}$

2. For each middle-seam $\overline{(s_i, s_j)}$
   if $convhull(\{B(s_i), B(s_j)\}) \not\subset P$ then
      (we just found a concavity)
      Let $B_{r_1}(p_1)$ and $B_{r_2}(p_2)$ such that
      p_1 and $p_2 \in \overline{(s_i, s_j)}$ and
      $d(p_1, p_2) = \max\limits_{convhull(\{B_{s_1}, B_{s_2}\}) \subset P)} d(s_1, s_2)$
      $Result = Result \cup \{B_{r_1}(p_1), B_{r_2}(p_2)\}$
      Mark $B_{r_1}(p_1)$ and $B_{r_2}(p_2)$ as articulation spheres;
3. For each End-Seam $E(s_i)$ do
      Let $B_r^1(s_i)$ be the maximal sphere centered
      on $E(s_i)$ and externally tangent to B_{s_i}
      $Result = Result \cup \{B_r^1(s_i)\}$
      Let $B_{r_2}^2(s_i)$ be the maximal sphere
      centered on $E(s_i)$ and externally
      tangent to $B_{r_1}^1(s_i)$ and such that $r_2 < r_1$.
      $Result = Result \cup \{B_{s_i}^2\} \cup \{B_{s_i}^2\}$

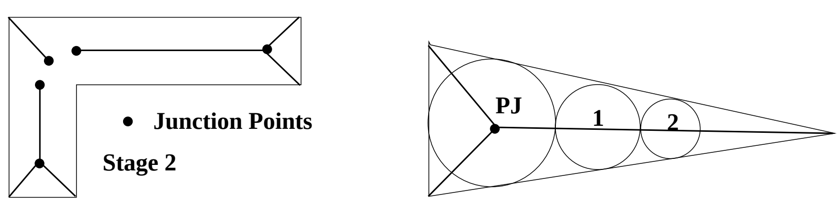


Figure 6: Steps of the algorithm

With the spheres selected by the previous algorithm, we construct the following graph called the *structure graph* (SG).

Definition 5 The structure graph $SG = (V, E)$ is a labeled graph such that:

V is the set of vertices called spheres. All the spheres are those computed by the Sphere Representation algorithm and belonging to the MAT,

E is the set of edges i.e. a binary relation defined by: $(x, y) \in E$ if x and y are related by a seam the MAT.

We have two functions defining labels on vertices and edges:

- L_s, which associates to each sphere the labels defining a type from (PJ, END, ART) and one or two sizing coefficients:
  - $(PJ, size)$ if the sphere is centered on a junction point, and the size is the ratio between the sphere and a reference one (the biggest in the MAT).
  - $(END, size, \rho)$ if the sphere has its center on a end-seam, $size$ is the ratio between the sphere and the reference one, and ρ the ratio of decreasing size on the end-seam.
  - $(ART, size)$ if the sphere has been marked as an articulation sphere.
- $L_e : E \longrightarrow \mathbb{R}^+$, which associates to each edge a distance (ratio) between the two spheres that define the edge. Note that in case of an edge between a sphere centered at a junction point and a sphere centered on an end-seam, the distance is null since they are tangent.

As a geometric graph, SG is a subset of the MAT since we consider all spheres at junction points, and the shape is uniquely represented by such a graph.

Moreover, we associate to this graph a function $D : V \rightarrow \mathbb{R}^3$, mapping a vertex with the coordinates of the center of the associated sphere that will be useful is some applications.

The resulting representation is simple as requested. We can observe that there is a sphere for each concavity of the object. This representation seems to be adequate to study the structure of an object.

Note that if the object is convex the previous algorithm reduces to points 1 and 3.

Example: The figure 7 shows a 2D convex form with the discs computed by 3.2 and the associated graph with labels. The labels of the vertices are the ratios wrt the biggest sphere.

3.3 Convex part graph and convex parts structure graph of the form

The structure graph of a concave shape can be complex. In order to have a more flexible representation, this graph can be splited in several parts. From

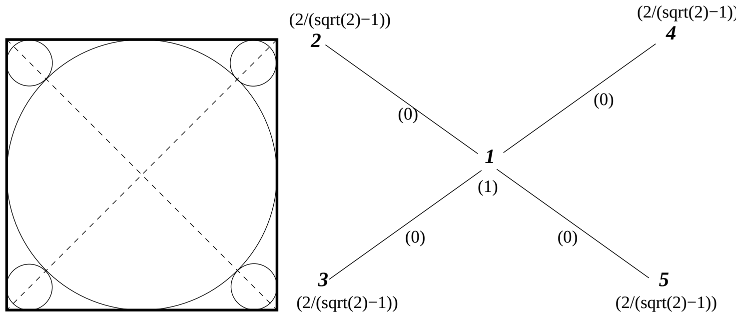


Figure 7: A convex form and its associated labeled structure graph

the structure graph is extracted a set of subgraphs called Convex Part Graphs (CPG) and corresponding to maximal convex parts. The convex hull of the set of spheres belonging to such a subgraph gives a rather accurate approximation of the shape of the part. Each sphere graph corresponding to a maximal convex part is delimited in the CPG by two sets of articulation spheres detected at stage 2 of the algorithm in section 3.2.

We now need to know the relations between the CPG, so we construct a new graph we call *Convex parts structural graph* $CPSG = (A, R)$, where:

- $A = \{p_1, \ldots, p_k\}$, where k is the number of maximal convex parts and each p_i is associated with a maximal convex part of SG,
- $\forall (p_i, p_j) \in A \times A, (p_i, p_j) \in R$ iff the associated convex parts P_i and P_j are such that: if A_i and A_j are the sets of articulation spheres of P_i and P_j, then A_i and A_j are connected by some edge in SG.

4 Construction of a colored graph

From the structure graph $SG = (V, E)$ (or CPG and CPSG) we construct a colored graph Γ, that is to say that we define colors on vertices and on edges: (i). Two vertices x and y will have the same color iff the spheres corresponding to x and y have the same size and, (ii) two edges will have the same color iff the distance between their corresponding extremity spheres is the same. Non-edges correspond to a color in order to complete the graph.

The equivalence classes of colors define a partition of the set V of vertices. This partition can be refined to obtains the partition into orbits of the automorphisms group of the graph using a so-called branch and bound method. This is the topic of the next section.

The construction of such a colored graph is necessary to compute automorphisms group with methods as Weissfeiler-Leman or Nauty, that are useful in some applications of the abstract framework described here ([Dugat *et al.*, 2002]).

5 Structural Analysis of a 3D form

One application of the structure graph is the detection of properties of the shape. Properties we will focus on here are structural invariance by isometries. We will call theses transformations *symmetries*. Symmetry detection in graph and geometric interpretation is a classic problem in graph drawing and is known to be NP-hard [Eades and Lin, 2000], [Hong, 2002]. Our problem is quite different: to detect symmetries that have a geometric sense in the form. The method we are about to expose will be used on structure graph. That is to say on the entire structure graph. But we can also modulate the granularity, i.e. in the case of a convex shape we can consider only the spheres of the first step of the algorithm in section 3.2 (those on the PJ points), to detect symmetries on the main part of the shape avoiding “details”. On concave shapes the computation can focus on CPG or CPSG to detect symmetries on the parts independently one of the other, or on the main relational structure of theses parts.

5.1 General results on symmetries

In order to understand what kind of problem we deal with here, we need some results on symmetries and graph structure.

Definition 6 [Abelson *et al.*, 2002] A *symmetry* α of a set of points Q in $\mathbb{R}^3$ is an isometry $\mathbb{R}^3 \to \mathbb{R}^3$ such that $\alpha(Q) = Q$.

As explained in [Hong, 2002], symmetries in the three dimensional space can by classified as *rotation*, *reflection*, *inversion* and *rotary reflection*:

- Rotational symmetry is a rotation about an axis ;
- Reflectional symmetry is a reflection in a plane ;
- Inversion is a reflection in a point ;
- Rotary reflection is a composition of a rotation and a reflection.

We only focus in this paper on rotations and reflections, i.e. the three first types of symmetries. If the shape is invariant by reflection or rotation then the center of the isometry can be materialized by vertices in the spheres graph or not.

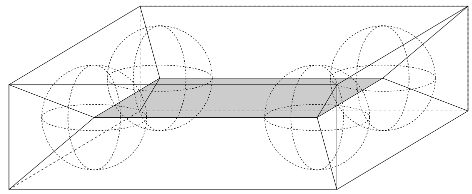


Figure 8: Center of symmetry.

For instance in fig. 8, the reflection with respect to the plane in gray is materialized by the four spheres. The reflections with respect to the other planes are not materialized.

It is well known ([Klin *et al.*, 1995]) that the group of automorphisms of a graph completely describe its structural (combinatorial) symmetries. We are only interested in automorphisms that are isometries and have a geometric sense, that is to say that the object is invariant in the 3D space using the orthogonal symmetry or the rotation defined by the automorphism. Of course the two notions are strongly related.

We establish the following propositions:

Proposition 3 The MAT is unique for a given form.

Proof: by construction □.

Proposition 4 The structure graph of a 3D form is a connected, non-oriented, simple and loopless graph.

Proof: by construction □.

Proposition 5 The structure graph is unique for a given form and granularity.

Proof: The spheres are at junction points of the MAT and along end seams. So the uniqueness of the MAT implies the uniqueness of the structure graph. The granularity will only affect the number of spheres along end seams □.

Proposition 6 A form is symmetric iff the structure graph is symmetric.

Proof: By construction it is obvious that a form is symmetric iff its MAT is symmetric. The structure graph is a substructure of the MAT that will keep the symmetries □.

An automorphism σ is reflectional if it is an involution, ($\sigma^2 = I$), that is to say if all its cycles are of order two, and the set of fixed vertices (order one) is either the empty set or a singleton (inversion), or defines a line or a plane (more then one singleton) ([Eades and Lin, 2000], [de Fraysseix, 1999]). In our case we must verify that this point to point mapping has a geometric sense using the coordinates of the centers of the spheres. This method will be called latter the "brute force" algorithm. Before we establish some useful properties:

An automorphism σ is a rotation ([Eades and Lin, 2000] if each cycle has order k, possibly except for several cycles with size one defining a line. A cycle C of σ with order k can be represented by $\{x, \sigma(x), \sigma^2(x), \ldots, \sigma^{k-1}(x)\}$ for any x of C.

Proposition 7 Let $\sigma \in Aut(SG)$. The cycles of σ form partitions which are a refinement of the orbits partition of the vertices of SG.

Proof: Let $(x_0, x_1, \ldots, x_k)$ be a cycle of σ. We have $\forall i \in \{0, k-1\}, x_{i+1 mod k} = \sigma(x_i)$. So all the elements of the cycle belong to the same orbit of $Aut(SG)$. Cycles are disjoint hence they define a refinement of the orbit partition of the set of vertices □.

Proposition 8 Let $\sigma \in Aut(SG)$ be a geometric symmetry. If there are spheres of SG belonging to the center of symmetry of σ then these spheres are fixed by σ.

Proof: Fixed elements are invariant by σ. So if it is a geometric symmetry, invariant elements must be on the center of symmetry □.

Proposition 9 Let σ be a geometric symmetry of $Aut(SG)$ then all pairs $(x, \sigma(x))$ of σ are of order two and their middle belongs to the center of symmetry of σ.

Proof: Obvious □.

5.2 The convex part case

Convex shapes are particular cases that have interesting computational properties:

Proposition 10 The structure graph of a 3D convex form is planar.

Proof: In the MAT, spheres are a measure of the thickness of the form, all in the same direction because of the convexity of the form and the maximality of the spheres. Moreover, when sheets and seams meet in the MAT we place a sphere, so the structure graph can be drawn without edges crossing and then is planar □.

Hong gives in [Hong, 2002] the following result:

Proposition 11 There is a polynomial time algorithm which computes a maximum size three dimensional symmetry group for planar graphs.

So in this special case combinatorial symmetries are computed in polynomial time for the structure graph $SG = (V, E)$. Each such symmetry is an automorphism σ mapping V on itself. For each couple $(v, \sigma(v))$ we must then verify the compatibility $D(v)$ and $D(\sigma(v))$ wrt the isometry corresponding to σ in the Euclidean space.

5.3 General 3D form case

In order to compute the orbits partition of the graph vertices, we use a branch and bound method as the Weisfeler-Leman algorithm ([Babel *et al.*, 1997]) or Brandan McKey's software Nauty based on a similar approach ([McKay, 2000]). Note that Nauty can also compute the complete group of automorphisms.

The above methods expect colored graphs, and to use them we interpret our labeled sphere graph as a colored graph as shown in section 4.

From the orbits partition of the set of vertices of the structure graph, we want to compute the automorphisms that are geometric isometries. We have several strategies: generate all the automorphisms with softwares are Nauty and select those that we want, or try to compute directly the interesting automorphisms. The next section compares the two strategies.

5.3.1 Brute force algorithm for general graphs

We shall show in this section that the brute-force algorithm can not be an efficient solution of computation. First we expose its principle:

1. Compute all the automorphisms of SG,
2. Keep in the previous set the automorphisms of order two (reflection) or $k > 2$ (rotation),
3. Keep in the previous set the automorphisms whose set of fixed points is empty or is composed of:
   - one vertex, rectilinear vertices or coplanar vertices in the case of the order two,
   - rectilinear vertices in the case of order $k > 2$,
4. In the case of a reflection, compute the middle of each cycle of σ, and verify with the coordinate function D that the middles are all rectilinear or coplanar and are compatible by an orthogonal symmetry whose center is the fixed points set if it is not empty.
5. In the case of a rotation compute the axis of the rotation of each cycle of σ and verify with D that the points are all rectilinear and are compatible with a rotation whose axis is the fixed points set if it is not empty.

Note: if the shape is convex the points 1 to 3 are replaced by: compute the maximal symmetry group

For sake of simplicity we only expose here the case of reflectional symmetry. Given a graph SG, the desired algorithm has to check for each automorphism σ

of $Aut(SG)$ whether σ corresponds to a geometric reflection of the object whose SG is the structure graph.

Let σ be an automorphism of $Aut(SG)$. σ represents a geometric reflection if and only if it verifies the three following properties (with $SG = (V, C, L_v, L_e, LV, LE)$ as defined in section 3.1):

1. $\sigma \in Aut(SG)$, i.e.:
   - $\forall a \in V \;\; L_v(\sigma(a)) = L_v(a)$
   - $\forall (a,b) \in V^2 \;\; L_e(\sigma(a), \sigma(b)) = L_e(a,b)$
   - $\forall (a,b) \in V^2 \;\; E(a,b) \leftrightarrow E(\sigma(a), \sigma(b))$
2. The order of σ is 2, i.e. σ is a reflection (thus not the identity function)
3. σ is a *geometric* reflection, i.e. $\forall (a,b) \in V^2$ such that $a \neq b$ ($\sigma \neq Id$ so it is always possible to find $(a,b) \in V^2$ such that $a \neq b$), $\sigma(a) = b$, $A = D(a)$ and $B = D(b)$ are the associated geometric points of the vertices a and b, P is the plane perpendicular to the line (AB) that intersects (AB) in M, middle of $[A, B]$, σ verifies *one* of the following properties:[5]

   Planar symmetry with regard to P:
   - $\forall (c,d) \in V^2$ if $c \neq d$ and $\sigma(c) = d$, then P is perpendicular to the line (CD) and intersects $[CD]$ in its middle .
   - $\forall e \in V / \; \sigma(e) = e$, then $E \in P$.

   Central symmetry with regard to M:
   - $\forall (c,d) \in V^2$ if $c \neq d$ and $\sigma(c) = d$, then M middle of $[CD]$.
   - $\forall e \in V$ if $\sigma(e) = e$, then $E = M$.

   Axial symmetry :
   - $\exists (c,d) \in V^2 - \{(a,b), (b,a)\} / \; c \neq d$ and $\sigma(c) = d$, N middle of $[CD]$, then $N \in P$.
   - $\forall (e,f) \in V^2 / \; e \neq f$, $\sigma(e) = f$, and O middle of $[EF]$, then $O \in (MN)$.
   - $\forall g \in V /$ if $\sigma(g) = g$ then $G \in (MN)$.

[5] For the sake of simplicity, if the lowercase letter "a" designs a vertex of SG, then the uppercase letter "$A = D(a)$" stands for the associated geometric point.

The main problem of this method is its computational complexity in the general case. For instance if the shape is a cube we have a sphere graph composed of one sphere centered on the unique junction point and eight spheres tangent with the previous one and located on end seams (if we stop algorithm of section 3.2 at step $2.j$ for $j = 1$). This graph with nine vertices has 40320 automorphisms. Only 13 are geometric symmetries: six are symmetries with respect to planes defined by opposite edges of the cube and three defined by the middles of parallel edges, three are symmetries with respect to lines defined by the center of opposite faces, one is a central symmetry. So it is more suitable to compute directly the automorphisms we are interested in, which can be done for the cube with theoretical algebraic graph tools since it is a convex shape, but not in the general concave case. So in the next section we present a branch and bound method to compute directly symmetric automorphisms.

5.3.2 Branch and bound algorithm for geometric symmetries research

We saw that the complexity of the naive algorithm (the above called "brute force") is exponential. To reduce the complexity, two different strategies can be envisaged:

1. Divide the research space.
2. For a given search space, try not to explore the whole associated search graph.

The first strategy is rather easy to apply: instead of doing the search for symmetries on the whole set of vertices of a graph SG, this can be done on each subset corresponding to the orbits. This property is directly derived from the definition of an orbit: considering an automorphism that would associate two vertices from two different orbits is useless, because the corresponding automorphism will not be a member of $Aut(SG)$, and thus not be a symmetry.

The second strategy is more complex: we gradually constrain the possible central element of symmetry. Given a central element and an orbit, we consider the permutation of two elements in this orbit. We check whether this permutation is compatible with the central element. The different possibilities are detailed in the graph in appendix A. The function realizing this evolution is the function *evo*: given a possible central element and an elementary permutation $x|y$ (x and y are possibly the same element), it returns the new possible central element, if any. This function is called recursively by SymOrbit which is applied on each orbit while there is still a possible central element. If a candidate element remains after examining all orbits, then a geometrical symmetry has been found.

Let $\{o_1, o_2, \ldots, o_n\}$ be the partition of Γ into orbits. The method is implemented by the following algorithm:

Notation 2 $x|y$ denotes a candidate permutation pair, l a list of vertices, GEL is a list of pairs. The first element is a candidate central element. The second element is the list of $x|y$ permutations already processed.

```
Main
    Result ← ∅(A global variable eventually
    containing the result)
    i ← 1
    GEL ← ∅
    Stop ← false
    do SymOrbit(Result, o_i)
    while ((Result ≠ ∅) and i < n)
    return Result
end

Comp(GEL, x|y)
     res ← ∅
     forall s ∈ GEL do
          s'.elt ← evo(s.elt, x|y)
          if s'.elt ≠ 0 then
            s'.σ ← s.σ ∪ {x|y}
            res ← res ∪ s'
     return res
end

SymOrbit(GEL, l)
    if l = 0 then
      Result ← Result ∪ GEL
    else
      forall b in match l = (a, reste) do
          GEL' ← comp(GEL, a|b)
          if GEL' ≠ 0 then
            SymOrbit(GEL', l − {a, b})
    return Result
end
```

6 Conclusion

We presented in this article a symbolic representation of a three-dimensional polyhedron which is a graph based on the medial axis transform. Using this graph we were able to detect structural (geometrical) symmetries of the object. Last we presented a branch and bound algorithm in order to reduce the computational complexity of the method in the general case. This is a first step for reasoning on abstract representations of objects and is an alternative to pure computational geometry or pure logical representations.

References

[Abelson *et al.*, 2002] David Abelson, Seok-Hee Hong, and Donald Taylor. A group-theoretic method for drawing graphs symmetrically. Technical Report Technical Report IT-IVG-2002-01, School of Information Technologies, University of Sydney, 2002.

[Babel *et al.*, 1997] L. Babel, I.V. Chuvaeva, M. Klin, and D.V. Pasechnik. Algebraic combinatorics in mathematical chemistry. methods and algorithms. ii. program implementation of the weisfeiler-leman algorithm, 1997.

[Berge, 1973] Claude Berge. *Graphs and hypergraphs*. North-Holland, 1973.

[Brand, 1992] J. Brand. Describing a solid with three-dimentional skeleton. In J. D. Warren, editor, *Proceeding of the International Society for Optical Engineering*, volume 1830, pages 258–269, 1992.

[de Fraysseix, 1999] Hubert de Fraysseix. An heuristic for graph symmetry detection. In *Graph Drawing, volume 1731 of Lecture Notes in Computer Science, Springer*, pages 276–285, 1999.

[Dugat *et al.*, 2002] Vincent Dugat, Pierre Gambarotto, and Yannick Larvor. Qualitative Geometry for Shape Recognition. In *Applied Intelligence*, volume 17, pages 253–263. Kluwer Academic Publishers, novembre 2002.

[Eades and Lin, 2000] Peter Eades and Xuemin Lin. Spring algorithms and symmetry. *Theoretical Computer Science*, 240(2):379–405, 2000.

[Godsil and Royle, 2001] Cris Godsil and Gordon Royle. *Algebraic Graph Theory*. Springer, 2001.

[Hong, 2002] Seok-Hee Hong. Drawing graphs symmetrically in three dimensions. In *Graph Drawing, 9th International Symposium, Vienna, Austria, September 2001, LNCS 2265*, pages 189–204. Springer, 2002.

[Hopcroft and Wong, 1974] J. Hopcroft and J. Wong. Linear time algorithm for isomorphism of planar graphs. *Sixth Annual ACM Symp. on Theory of Computing*, pages 172–184, 1974.

[Klin *et al.*, 1995] M. Klin, C. Rucker, G. Rucker, and G. Tinhofer. Algebraic combinatorics in mathematical chemistry, 1995.

[Macrini *et al.*, 2002] Diego Macrini, Ali Shokoufandeh, Sven Dickinson, Kaleem Siddiqi, and Steven Zucker. View-based 3-d object recognition using shock graphs. In *16 th International Conference on Pattern Recognition (ICPR'02), Quebec City, Canada*, volume 3, 2002.

[McKay, 2000] Brendan McKay. *Nauty user's manual*. Australian National University, Cambera, 2000.

[Sherbrooke *et al.*, 1996] Evan Sherbrooke, Nicholas Patrikalakis, and Erik Brisson. An algorithm for the medial axis transform of 3d polyhedral solids. *IEEE Trans. on Visualisation and Comp. Graphics*, 2(1):44–61, 1996.

[Sherbrooke, 1995] Evan Sherbrooke. *3D Shape Interrogation by Medial Axis Transform*. PhD thesis, MIT, Cambridge, Massachusets, 1995.

A Annex: Evolution function

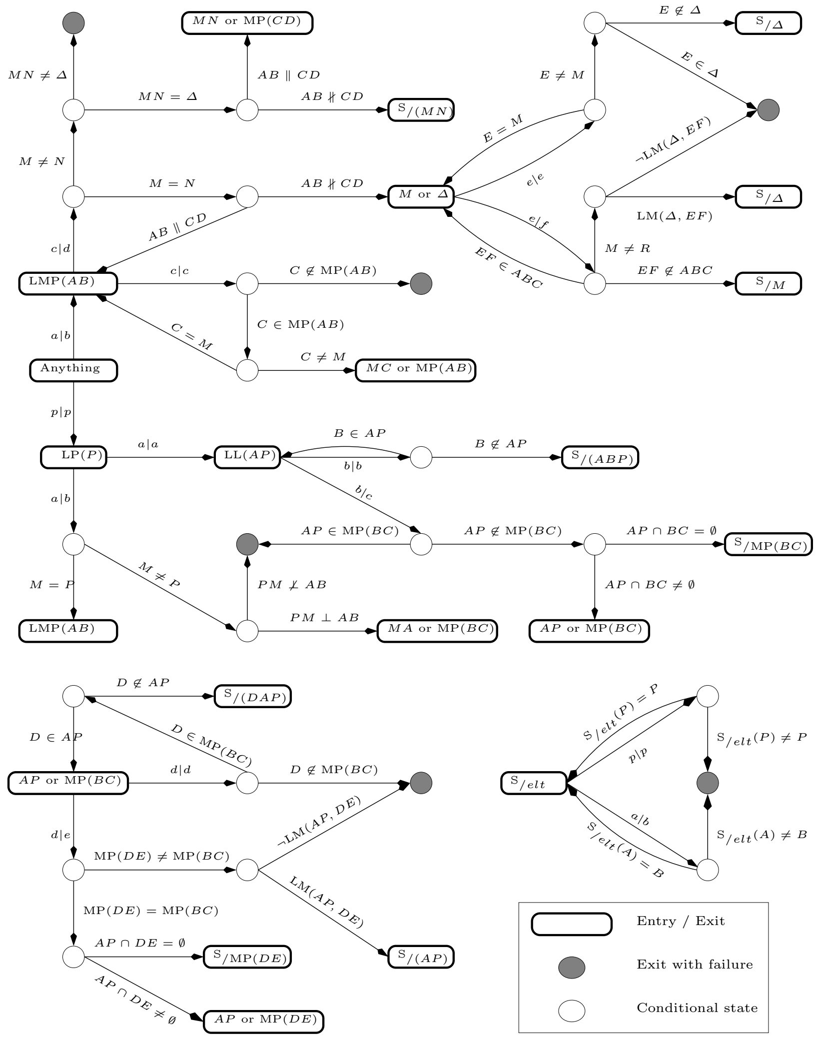
We present the principle of the Branch and Bound algorithm as a graphic which resumes the different steps. How to read the graph on next page:

- Vertices in thick rounded box are entry and/or exit points, depending on whether the arcs lead from and/or to them. They bear as labels the geometrical element given to/returned by the function emphevo. The arcs leaving these nodes are only labeled with the pair we try to add to the current automorphism.

  The vertices labeled $S_{/elt}$ correspond to the case when we have found a definite reflectional element elt, be it a plane, line or point.

- Round grey vertices mark a dead end: the tested configuration is impossible, *evo* returns a void central element.

- Round blank vertices mark testing points. Arcs stemming from them bear mutually exclusive conditions, which are tests on the possible geometric configurations at these points in the graph.



MP(XY) = mediator plan of X and Y
LM(L, XY)=true iff L is a medriatrix of X and Y
Δ= common mediatrix of AB and CD

M = midpoint of A and B
N = midpoint of C and D
R = midpoint of E and F

LMP(AB) = MP(AB) or Δ or M
LP(P)= P or any line or plan through P
LL(AP)=any line or plan through A and P