

Fast global and partial reflective symmetry analyses using boundary surfaces of mechanical components

Ke Li, Gilles Foucault, Jean-Claude Léon, Moreno Trlin

▶ **To cite this version:**

Ke Li, Gilles Foucault, Jean-Claude Léon, Moreno Trlin. Fast global and partial reflective symmetry analyses using boundary surfaces of mechanical components. Computer-Aided Design, 2014, 53, pp.70-89. 10.1016/j.cad.2014.03.005 . hal-00961709

HAL Id: hal-00961709

https://hal.univ-grenoble-alpes.fr/hal-00961709

Submitted on 20 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*Highlights (for review)

Fast global and partial reflective symmetry analyses using boundary surfaces of mechanical components

K. LI[a], G. FOUCAULT[a,*], J-C. LEON[b], M. TRLIN[b]

[a] *University of Grenoble - G-SCOP laboratory, 46 av. Félix Viallet, 38000, Grenoble, France*

[b] *University of Grenoble / INRIA - LJK laboratory, 655, av. de l'Europe, 38334 Montbonnot, France*

Abstract

Axisymmetry and planar reflective symmetry properties of mechanical components can be used throughout a product development process to restructure the modeling process of a component, simplify the computation of tool path trajectories, assembly trajectories, etc. To this end, the restructured geometric model of such components must be at least as accurate as the manufacturing processes used to produce them, likewise their symmetry properties must be extracted with the same level of accuracy to preserve the accuracy of their geometric model. The proposed symmetry analysis is performed on a B-Rep CAD model through a divide-and-conquer approach over the boundary of a component with faces as atomic entities. As a result, it is possible to identify rapidly all global symmetry planes and axisymmetry as well as local symmetries. Also, the corresponding algorithm is fast enough to be inserted in CAD/CAM operators as part of interactive modeling processes,

*Corresponding author: tel. +33 476574310

Email addresses: Gilles.Foucault@ujf-grenoble.fr (G. FOUCAULT), Jean-Claude.Leon@grenoble-inp.fr (J-C. LEON), Moreno.Trlin@inria.fr (M. TRLIN)

it performs at the same level of tolerance than geometric modelers and it is independent of the face and edge parameterizations.
Keywords:
reflective symmetry, axisymmetry, B-Rep CAD model, accuracy,
CAD/CAM, divide-and-conquer, local symmetry

1. Introduction

The detection of shape symmetry properties (both axisymmetry and planar reflective symmetry) finds a large range of applications and covers numerous categories of geometric models. The use of symmetry properties to characterize shape similarities through a shape descriptor has produced numerous contributions [1, 2, 3, 4, 5, 6]. Shape symmetry properties are also helpful to highlight shape structures [7, 8]. Symmetry can contribute also to the compression of shape models [9].

Here, the context is placed on mechanical components, as described through B-Rep CAD models. The corresponding shape models addressed are solids and their symmetry properties are useful to restructure their feature tree (see Figure 1). Indeed, when a designer generates a solid, often its feature tree does not contain the mirror operations or other symmetry constraints, because a design process includes functional objectives and modifications that prevent the designer from generating a well structured feature tree of the final solid (see Figure 1b). Therefore, analyzing the local as well as the global symmetry properties of a solid is useful to reorganize its feature tree so that successive modifications can take advantage of its effective symmetries and become more intuitive (see Figure 1c). The object model thus obtained

is often designated as a structured and/or compressed model. It has to be pointed out that the location of symmetry planes and the dimensions of the symmetric primitives obtained through the symmetry analysis must produce a compressed model that replicates the input model within the accuracy of the manufacturing process, at least. Let M, M', T and d designate respectively the initial object, the compressed one, the geometric transformation expressing the symmetry and a distance measure. Then, the replication error of M with M', homologically identical to M, and T can be measured by the Hausdorff distance:

$$\epsilon = d(M, T(M')) = \max\left(\sup_{\mathbf{p}\in M}\left(\inf_{\mathbf{p}'\in M'}||\mathbf{p}' - \mathbf{p}|\,|\right), \sup_{\mathbf{p}'\in M'}\left(\inf_{\mathbf{p}\in M}||\mathbf{p}' - \mathbf{p}|\,|\right)\right)$$

Now, let δ be the CAD modeler tolerance expressing the maximal distance between boundaries of adjacent patches. δ is mandatory to process shapes incorporating trimmed patches. Then, $\epsilon \leq \delta$ is the condition of validity of M' so that M and M' can be regarded as identical in the CAD modeler. Consequently, conventional machine tools used to manufacture M' produce an object that cannot be distinguished from the object manufactured from M. This means that the object boundary sampling process of M, often part of symmetry analyses [6], must be either dense enough so that the validity condition is satisfied with M' or global properties of surfaces bounding M must be combined with the sampling process to reduce the density of the latter [10].

Analyzing the symmetries of a component is also useful at other steps than design. As an example, component symmetries can be taken into account to compute tool path trajectories of a machining process and structure

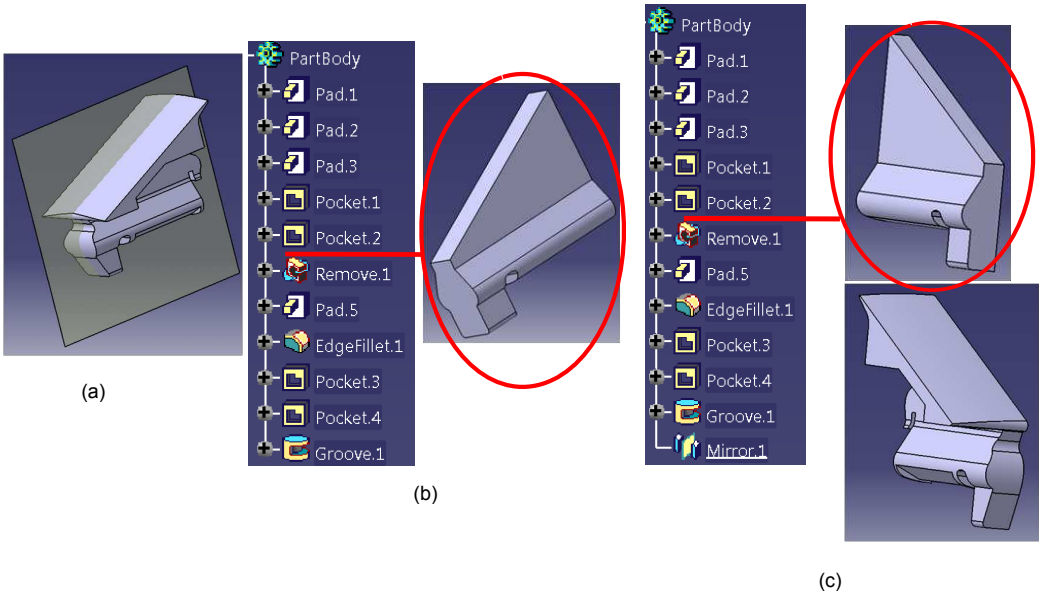


Figure 1: An example of construction tree reorganization based on symmetry properties. a) an object with a global symmetry plane, b) its corresponding construction tree and one of its intermediate steps, c) its construction tree after taking into account its symmetry properties and the corresponding modification of some of its primitives.

these trajectories to optimize the tool displacements. Similarly, symmetry properties of components contribute to that of assemblies and help defining trajectories that optimize assembly processes. In topological optimization, the automatic generation of new designs takes advantage of symmetries defined on the design and non-design sub-domains in the initial CAD model [11]. Finally, symmetries of components combined with the symmetry of boundary conditions form the basis of possible simplifications of the domain used in finite element models since a symmetric configuration forms the basis of a domain reduction expressing the symmetry of a physical phenomenon. The above configurations are common ones that are often appearing during a product development process. However, the symmetries engineers commonly refer to, may not be strict geometric properties (see Figure 2) and show how the concept of symmetry can evolve and get designated as repetitions. Additionally, symmetries of interest are not those valid for the entire object but also some that are valid on a subset of the object boundary, i.e. partial ones [8].

The above configurations share a common requirement: the geometric differences between M and M' must preserve the validity condition stated above. Consequently, if symmetry properties are incorporated into tool path trajectories, they produce a machined component that is as accurate as the machined component obtained without using its symmetry properties. Assembly and finite element simulation processes share also the same requirements. The validity condition can be expressed as a need to obtain symmetry properties as accurate as the modeling kernel of a CAD system since the incorporation of symmetry properties in the feature tree or in the machining

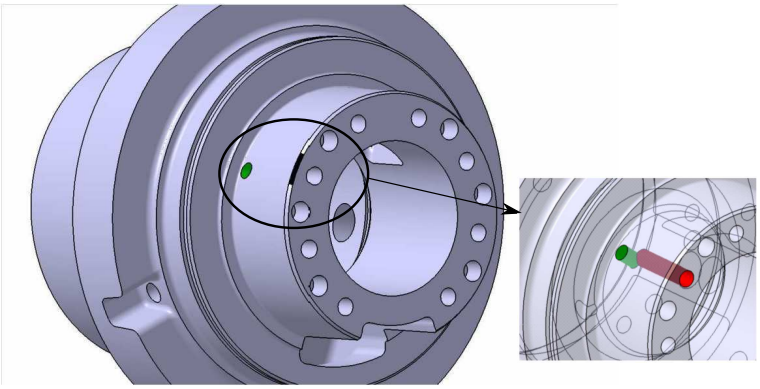


Figure 2: Mechanical component featuring a local, not exact, rotational symmetry of holes. The detail image shows that the rotational symmetry is influenced by an orthogonal hole (in green) crossing an element of this symmetry (in red) (courtesy ANTECIM).

trajectories of M must produce M' that does not deviate from M with a distance greater than the accuracy δ of this CAD modeling kernel.

From a complementary point of view, to integrate symmetry analysis in design and manufacture processes, it is also necessary to perform this analysis in an interactive time. This is particularly true in the design context where all modeling operations must be performed interactively to let the engineer generate efficiently components. Consequently, restructuring a feature tree under symmetry constraints is an operator belonging also to the modeling process and hence, it must be interactive.

It is the purpose of the proposed approach to meet both the accuracy and interactivity requirements. Section 2 analyzes prior work to put forward their key features, section 3 states the principles and hypotheses of the approach, section 4 describes the mandatory boundary decomposition transformations to obtain a representation intrinsic to the symmetry properties of the component. Then, sections 5 and 6 detail the divide and conquer phases of the

symmetry analysis process. Section 7 illustrates the results and performances currently obtained.

2. Related Work

Based on the major requirements derived from the previous section, prior work is reviewed with a particular focus on the validity condition. Previous approaches differ according with the categories of symmetries detected, whether these symmetries are detected exactly or approximately, globally and locally. They differ also in the type of geometric model processed as well as in the objectives of the symmetry detection. The type of geometric model has a direct incidence over the sampling process mentioned in the previous section because if prior work uses a geometric model differing from B-Rep NURBS, a model conversion may be necessary to process CAD models, hence a potential sampling process. The major features of the symmetry detection methods previously proposed relate to global vs. partial, exact vs. approximate, intrinsic vs. extrinsic symmetries [6]. Alternatively, prior work can be structured according to the category of entity used to detect symmetric configurations, i.e. points, curves or polylines, surface segments, ...

Exact and global symmetries of point sets and polyhedrons have been extensively studied showing that detecting symmetries for 3D point sets and polyhedrons can be performed in $O(n \log n)$ time [12, 13, 14] (n is the number of points). This low complexity is attractive but if applied to a B-Rep CAD model with smooth boundary it requires a very dense sampling to obtain a measure of symmetries and reach the validity condition. Typically, the tolerance δ ranges between 10^{-3} to 10^{-5} of unit length in commercial

CAD software for objects having maximal dimensions of 10^3 unit length, i.e. the validity condition fall within 10^{-6} to 10^{-8} range. Then, if a B-Rep CAD object boundary is converted into a point set without referring to the underlying type of surface, i.e. plane, cylinder, ..., and to its parameterization (distance between sampled points as well as the location of a reference point), its discretized representation would require millions of points to obtain a valid symmetry analysis within the tolerance δ. Points alone are not suited to reach the validity condition and higher order geometric entities must be processed somehow.

Focusing on triangle meshes, approaches have been proposed that take advantage of more global geometric information. Generalized moments were used in [15] and combined with mesh segmentation to reduce, as much as possible, the order of the moments. Even though spherical harmonics help evaluating moments, a quadratic number of surface integrals is needed, which significantly increases the algorithm complexity. The work of Mitra [9, 16] takes advantage of discrete curvature analysis and clustering to reduce the complexity of the input mesh ; pruning nearly umbilic areas to avoid ambiguous comparisons. The reduced n-sized point set thus obtained leads to an $O(n \log n)$ time complexity but small features falling in between the key points retained are not taken into account in the symmetry detection. Such a filtering behavior is not applicable to the engineering applications addressed here. Additionally, mesh segmentation is hardly a robust process, which makes the clustering process sensitive to the mesh connectivity and inadequate for engineering applications. Because these approaches process meshes, analyzing a B-Rep CAD model would require to discretize its

boundary. Again, if this sampling process if it does not refer to the underlying surface types, it is still reflecting their parameterizations, at least by the location of a reference point, requiring too large mesh size to reach the validity condition. More recently, feature lines have been used [17] to set a curve network matching problem but this does not solve the issue regarding the small features.

Approximate symmetry detection has been also addressed in the context of reverse engineered objects [18] to detect congruences in objects. Reverse engineering objects strongly couples data denoising, segmentation, occlusion processing with geometry processing, which differentiates this context from the CAx models addressed here. Computing approximate symmetries can be an NP-hard problem as pointed out by Iwanowski [19]. Referring to the underlying canonical surfaces attached to point subsets, e.g. planes, cylinders, cones, spheres, tori, this complexity has been reduced by Mills et al's approach to a low polynomial time in between $O(n^2)$ and $O(n^{4.5})$ when model features are available after a reconstruction process [10]. There, approximation of congruences are inherent to the shape acquisition process. Indeed, canonical surfaces can be characterized by key points, which drastically reduces the amount of points to be processed in congruence detection compared to the amount of digitized points. Because of the initial digitized model, the location of these key points along and inside the faces boundaries are subjected to inaccuracies inherited from the digitization process as well as influences from the edge and face parameterizations where their origin and directions may differ for paired faces and edges. However, the latter can be regarded as a similar order of magnitude as the digitized ones. Later,

further work focused on identifying local approximate symmetries [20] with a complexity of $O(n^4)$. Recently, the decomposition of the B-Rep model into a Regularity Feature Tree (RFT) has been combined with the detection of local approximate symmetries [21, 8] as a means to connect the symmetry detection with a shape structure. This reduces the amount of symmetries being processed and produces more meaningful entities to structure a B-Rep model. In [22], exact symmetries are derived from the feature tree of the object. This analysis is restricted to a CAD modeler context since feature trees are not exported to STEP or other standard format. In addition, some modeling hypotheses restrict the scope of the approach to categories of feature trees without redundancies. However, some of these redundancies can be difficult to avoid, as it is the case for the symmetry itself that could be embedded in the object's feature tree. Additionally, the construction tree is not updated in accordance with the symmetries identified. In the large, feature trees or construction trees (solid decompositions) have been addressed in different contexts [23, 24, 25] but they are lacking the use of symmetry properties.

Indeed, the approach proposed by Li et al. [26, 20, 8] detects symmetries from point sets derived from the surface boundary description of a B-Rep. Converting the faces and edges of a B-Rep model into a point set is sensitive to their parameterization. Parameterization or reparameterization of edges can be addressed with dedicated algorithms [27, 28, 29, 30, 31] but closed curves add the problem of defining appropriately their origin (see Figure 3), which stays an open problem. On Figure 3, intersection curves between the cylinders and the sphere are spatial closed curves P_1, P_2, P_3. The set of open

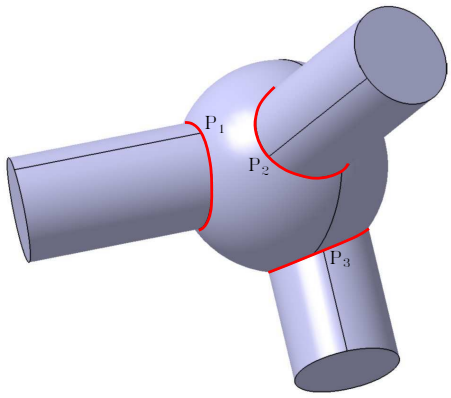


Figure 3: Component featuring a set of canonical surfaces interacting through spatial closed curves. Here, cylinders' axes do not contain the sphere center.

curves forming each P_i may not be located symmetrically with respect to each other, even though the cylinders and sphere are symmetrically set up. Indeed, the intersection curve should be a closed curve but setting up its origin that would be intrinsic is complex. Referring to singularities of surface intersections would require some classification [29]. Reparameterization approaches require some curve pairing [27, 30], which is already an output of a symmetry analysis and necessitates a first match based on their extreme points, for example. Likewise, different face parameterizations become difficult to align. These parameterization issues are based on optimization techniques, hence they are time consuming. It is often difficult to tune their control parameters to keep the results within the accuracy of a modeling kernel. A more generic approach would be to avoid referring to curve and surface parameterizations, which is part of the contribution described hereafter.

Also, the computation of congruences requires a combinatorial approach whereas the usages mentioned in a product development process require essentially reflective symmetry and rotational symmetry into some extent. Other transformations are useful for shape analysis purposes but they are not part of the modeling operators currently available in commercial software and rotational symmetry in mechanical components contains technological meaning that is worth a specific algorithm (see Figure 2).

Another approach, aiming at detecting symmetries from B-Rep CAD models uses global indicators with face centroids and a loop pairing algorithm to identify symmetry planes and rotational symmetries [32]. However, the corresponding algorithm relies on heuristics and cannot be considered as robust and the time required to evaluate accurately the centroids leaves it also out of the interactive time required to incorporate it into modeling operators.

3. Overview and hypotheses

3.1. Symmetry analysis and shapes

Here, the purpose is the detection of partial and global reflective symmetries as well as the partial and global axisymmetries of a B-Rep CAD model. This model is bounded by a 2-manifold. The corresponding symmetry planes Π_i and axes A_j must satisfy the validity condition (see section 1), i.e. Π_i and A_j must be identified within the accuracy of the modeling kernel of CAD software so that the symmetry properties of an object can be effectively used during design and simulation processes. Partial symmetries are also of interest to structure a modeling process. In this case, modifications in a feature

tree should combine, at least, variants of feature trees containing CAD modeling operators, e.g. extrude or cut operators, and symmetry properties to derive a new construction tree.

The proposed approach concentrates on a symmetry analysis able to process the widest diversity of shapes. Global symmetry properties are mandatory to restructure feature or construction trees. Indeed, global symmetry properties are meaningful after each step s_i of a modeling process, in a backward manner. In Figure 1a the global symmetry plane can be used to restructure the construction tree like in Figure 1c. Figure 1a can be also regarded as a final result of an intermediate step s_{i-1} of a modeling process (see Figure 4), hence the interest of global symmetry properties. Local symmetry properties are useful to analyze, at each step s_i of a modeling process, how primitives are attached to the result of its prior step s_{i-1} (see Figure 4).

The proposed approach is a first step to fit into the above context where the objects are effectively bounded by planar, cylindrical, conical, spherical, toroidal faces. This is originated by the modeling process of mechanical components using current CAD software. Most of the time, engineers model components from 2D sketches containing essentially line segments and circles or arcs and the content of these sketches is used to generate solid primitives through orthogonal extrusion or revolution operators. This is also confirmed from a complementary point of view through a shape study in [33]. The input model could be acquired as:

- a B-Rep CAD datastructure if the symmetry analysis is an operator tightly connected to a modeling kernel;
- a B-Rep CAD model imported using a neutral file format if the sym-

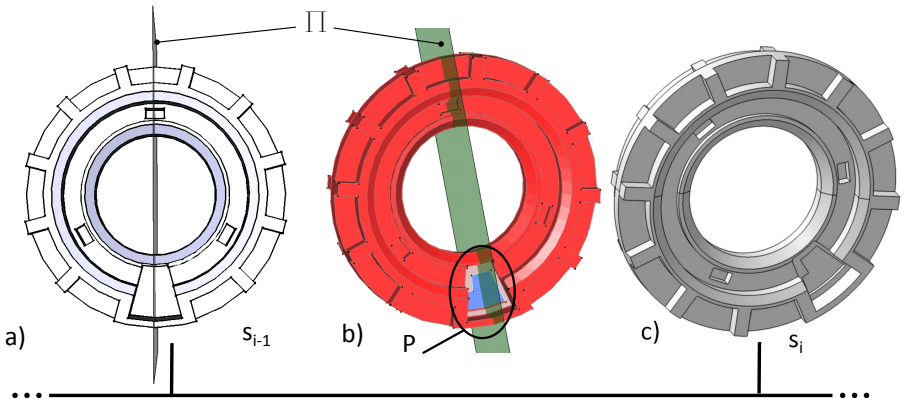


Figure 4: a) An example of local symmetry plane Π and its relationship with the attachment of a primitive shape P of pocket type at a step s_i of a modeling process. The blue face belongs to the primitive shape. Blue and pink faces contribute to the attachment of P and are not symmetric with respect to Π. b) Red faces are symmetric with respect to Π. At s_i, Π is a local symmetry plane but going backward in time, Π becomes a global symmetry plane at s_{i-1} showing that the attachment of P is a symmetry breaking operation in the construction tree of the object. c) Result at s_i with no global symmetry plane.

metry analysis takes place as a library component outside a modeling kernel.

Assuming that the neutral format is STEP, both configurations provide the B-Rep CAD description with faces categorized under the five required types.

3.2. Principle of the symmetry analysis using a divide-and-conquer approach

As a synthesis of prior work, it appears also that symmetry analysis can be sped up when complementary geometric information can be added to the input point set, e.g. centroids, moments, discrete curvatures, etc. or higher level entities are used. The current approach uses the fact that any Global Symmetry Plane (GSP) of a solid M defined with one connected component must intersect this volume (see Figure 6). A Global Symmetry Axis (GSA) is derived as a particular set of GSPs, i.e. an infinite set of GSPs sharing

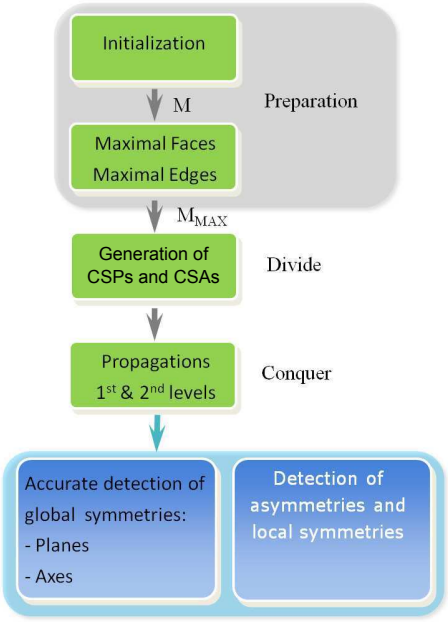


Figure 5: An overview of the symmetry analysis through the main steps of the algorithm.

a common intersection line. The global scheme of the proposed approach is depicted on Figure 5. The preparation phase from a solid M and its boundary ∂M aims at generating a new paving of ∂M that is intrinsic to the shape of M. This transformation is strictly topological and produces M_{MAX} with its paving ∂M_{MAX} containing maximal faces and edges. The description of criteria and key transformations to obtain M_{MAX} is given at Section 4. Then, the purpose of the divide phase is to take advantage of the basic configurations between symmetry planes and ∂M_{MAX} to assign Candidate Symmetry Planes (CSPs) and Candidate Symmetry Axes (CSAs) to vertices, edges and faces of ∂M_{MAX} so that each CSP or CSA is locally valid for a set of one, two or three adjacent faces depending on configurations described at Section 5.

Next, comes the conquer phase of the analysis that is further subdivided into two propagation processes. The principle of the first one can be summarized as follows: it starts from an arbitrary CSP or CSA, Π, obtained from the division phase and identifies the corresponding entities in ∂M_{MAX} and propagates to neighboring entities of ∂M_{MAX} when one of their associated CSPs coincide with Π, thus expanding the area of ∂M_{MAX} where Π is valid (see Section 6.1). Successive CSPs or CSAs that coincide form a CSP chain, i.e. a set of adjacent faces of ∂M_{MAX} where Π is valid, and this propagation stops when either a CSP chain forms a loop, i.e. the set of adjacent faces attached to Π forms a cycle over ∂M_{MAX}, or the coincidence criterion fails. The latter configuration characterizes an asymmetric configuration. Closed CSP chains reflect the intersection between a CSP and M_{MAX} to show that there exists a symmetric neighborhood in M_{MAX} around this CSP. Each such loop is a 1-cycle [34] reflecting the topology of M. Then, CSP chains that belong to coinciding CSPs are aggregated to form a single set describing the symmetric area of M_{MAX} around Π.

The principle of the second propagation level starts from a CSP chain and evolves from face to face, on both sides of this CSP chain, through adjacency relationships described in the paving of ∂M_{MAX} (see Figure 6 and Section 6.2). Its purpose is to check that two couples of faces, each one located on each side of a CSP chain, are symmetrically set with respect to Π, the CSP originating the chain. This process stops either when all the faces of M_{MAX} have been processed or when asymmetric configurations have been encountered everywhere along the domain boundary expanded from a CSP chain.

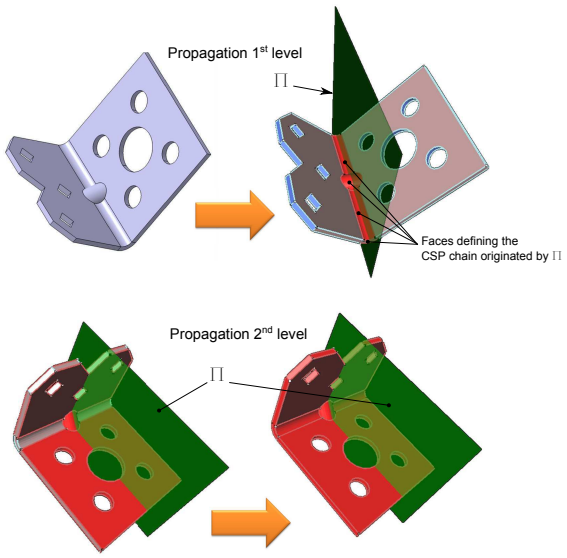


Figure 6: Illustration of the 1^{st} and 2^{nd} levels propagation processes. The 1^{st} level starts from a CSP, Π, obtained from the division phase and propagates around ∂M_{MAX} through neighboring CSPs when they coincide (initial object on the left). Faces and edges contributing to the propagation are colored in red when they are symmetric with respect to Π. Faces having an underlying surface symmetric and at least one of its boundary edges not symmetric are colored in pink. Unvisited faces and edges are colored in blue. The 2^{nd} level starts from faces and edges of a CSP chain (on the left) and propagates from both sides of Π. All red faces and edges indicate that Π is a GSP in the present example.

At the end of this conquer phase, if all the faces of M_{MAX} have been covered after the second propagation, the CSP is indeed a GSP. When a propagation stops earlier than the complete coverage of M_{MAX}, symmetry of M_{MAX} around the corresponding CSP is delimited by the areas of M_{MAX} covered by the propagation processes. These areas define local symmetry properties of M_{MAX} based on the adjacency of faces in M_{MAX}. This simplified description of the divide-and-conquer process illustrates the principle of the proposed approach and shows how global as well as local reflective symmetry properties of M can be obtained.

Regarding axisymmetry, the same process of division and conquer pro-

cesses apply to assign CSAs to edges and faces of ∂M_{MAX} and then, propagates this axisymmetry property along a chain until it either covers entirely ∂M_{MAX} and forms a GSA or stops and defines the locally axisymmetric area of ∂M_{MAX}.

4. Generating the boundary of M_{MAX}

Indeed, it is mandatory that ∂M paving does not influence the symmetry properties of M by its decomposition into faces, edges and vertices. This paving must be intrinsically linked to the solid's shape to meet the validity condition. Whether M is imported as a STEP file or directly accessed from a modeling kernel, ∂M reflects its modeling process and its boundary decomposition may not be symmetric even though the shape of M owns GSPs. Figure 7 exemplifies decompositions influencing the symmetry analysis. In Figure 7a, M is a symmetric shape but ∂M paving is not whereas Figure 7b is. Here, the non symmetric decomposition can be seen as the result of a user's modeling process because his, resp. her, modeling operations don't preserve the shape symmetry. In Figure 7c, M is an axisymmetric shape but ∂M paving is not, whereas Figure 7d is because it contains one face and no vertex along boundary edges. The non axisymmetric decomposition derives from topological requirements of modeling kernels to describe a solid.

It has to be pointed out that approximate symmetry detection based on meshes is not subjected to this constraint whereas the loop comparison process in [32] ought to take into account these decomposition issues to meet the validity condition. In [10, 20], the authors generate a point set from ∂M and use the concept of compatible faces. It is not referred to merge faces

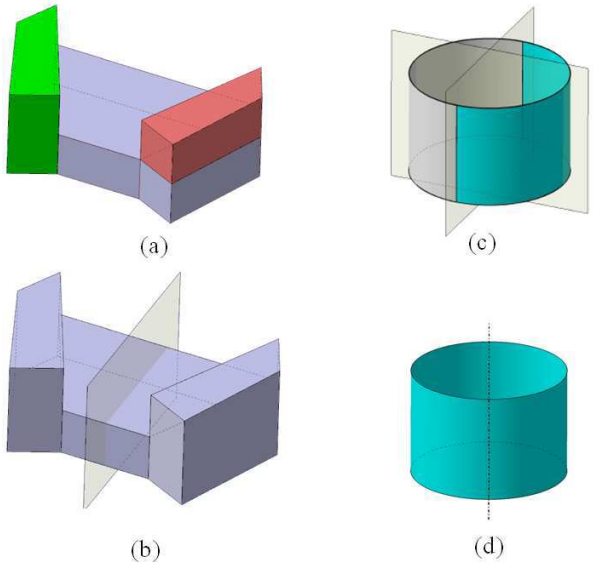


Figure 7: Examples of boundary decompositions influencing the symmetry analysis. (a) a symmetric shape with a non symmetric decomposition, (b) the same shape with a symmetric decomposition, (c) a symmetric decomposition without axisymmetry, (d) a decomposition producing axisymmetry.

and/or edges operators that could modify the connectivity of vertices, change the number of edges per loop. Starting from a ∂M paving that is subjected to modeling process issues and modeling kernel constraints stay sensitive to these phenomena. As a example, any arbitrary boundary decomposition of a sphere as generated by modeling kernels would require paving transformations to produce a new paving containing a single face with no edge and no vertex to be independent of modeling processes, modeling kernel constraints, face and edge parameterizations, origin of parameterizations. In [22], the feature interactions that cannot be processed solely with feature parameters may rely on [20]. Indirectly, this feature-based approach is also sensitive to the paving of ∂M.

Here, the preparation phase takes care of the transformation of ∂M paving to configure it so that it does not change the shape of M, i.e. only topological transformations are performed, but produces a new paving that is really intrinsic to the shape of M. This can fit into the concept of virtual topology in the large. However, criteria used in [35, 36, 37] can merge faces of different types and commercial software implementing virtual topology operators may incorporate geometric transformations and approximations as well. It is the purpose of this section to state the specific criteria and operators needed for symmetry analysis.

The transformation of ∂M uses concepts of maximal faces and edges and the resulting paving is noted ∂M_{MAX}. Hence, the designation of the object becomes M_{MAX}. Maximal faces were studied in [38] as c-faces to define ∂M as a Boolean combination of half-spaces. No shape transformation takes place during this phase, hence all symmetry properties are extractable from

∂M_{MAX} and it is the purpose of the transformation from ∂M to ∂M_{MAX} to generate a shape segmentation, i.e. a boundary decomposition, that preserves the symmetry properties of a solid. Transformations address adjacencies between faces, edges and vertices only, there is no modification of surfaces that define the shape of M. The topological datastructure describing the paving of ∂M_{MAX} is built on top of the topological datastructure of the B-Rep. The specific needs for the symmetry property preservation can be listed as follows:

a. Two faces F and F', adjacent to each other along a common edge, that describe the same underlying surface S must be merged. This requirement produces the so-called 'maximal' faces. If $S_F(x, y, z) = 0$ and $S_{F'}(x, y, z) = 0$ are the equations of F and F', respectively, it means that $\forall(x, y, z),\ S_F \equiv S_{F'} \equiv S$. Indeed, a maximal face is obtained once the face merging can no longer be applied. Typically, faces can be merged using a tolerance ϵ of the same value of that of a modeling kernel. Then, the resulting face bounds a half-space and, according to Silva's [38] definition, it is a 'c-face'. Silva proved that c-faces uniquely define the boundary of an object. Consequently, these faces are intrinsic to the object and independent of any modeling process. It can be observed that closed surfaces reducing to a single reference surface produces a single face without an edge and a vertex. This consequence applies to spheres and tori, forming faces without edges and vertices;

b. Two edges E and E', adjacent to each other at a vertex where they are the only connected edges, can be merged. Indeed, such a configuration indicates that the faces on each side of E and E' are identical. Hence, E

and E' can be merged to form a so-called 'maximal' edge. Therefore, a maximal edge is uniquely defined by its adjacent maximal faces and the set of maximal edges is unique for ∂M_{MAX}. Generating maximal edges is a purely topological operator based on the maximal faces. Similarly to faces, closed edges can be merged up to be bounded by one or no vertex at all. Circles are examples of closed maximal edges not bounded by vertices. However, these face and edge definitions do not provide a convenient way to describe ∂M_{MAX} using constructive processes, as pointed by Requicha [39]. This justifies the use of hypergraphs as a topological datastructure describing ∂M_{MAX} on top of ∂M;

c. Another configuration relates to the loops forming the boundary decomposition of M. Loops can be adjacent to each other through edges or vertices only. Indeed, adjacency through edges is already covered by configurations [a]. Here, the only configurations addressed are loops adjacent at a vertex. When a vertex V is connected to more than two edges, the above edge merge operator is not applicable. In order to obtain meaningful local symmetry information, it is important to precisely delimit valid areas so that they can be used in connection with modeling features during the generation of construction trees or construction graphs [40] (see Figure 8). Bounding the symmetry property of faces with their closure, i.e. their edge loops, is therefore an effective need to separate shape features as much as possible. The corresponding configurations are analyzed and a corresponding operator is briefly described hereafter.

Vertex V_1 on Figure 8a gives an example of a configuration [c] with edges

E_2 and E_3 of loop L_2. Locally, F_3 has an axisymmetry property still valid along E_2 and E_3 but V_1 influences the local symmetry properties of F_3 and its boundary (see Figure 8c). The purpose of this transformation is to characterize the independence of surfaces around a vertex so that local symmetry properties can be better represented if loop connections are removed by splitting their common vertex (see Figure 8d). This transformation has no impact on the global symmetry properties of M, if any, since a symmetry plane Π either contains V or another vertex V' exists that is symmetric to V with respect to Π. Somehow, F_3 becomes independent from the other faces around V_1 when a vertex V_1' is created by splitting into V_1 and V_1'. The couple of face sectors (F_2, F_2) is now merged and creates a new face F_i' (see Figure 8b top). As a result, symmetry properties of F_3 can be recovered because E_2 and E_3 bounding F_3 now form a maximal edge E_i' and F_3 has some independence with respect to its neighboring faces once V_1 is split (see Figure 8b top and bottom and 8d). This vertex split reasoning can then be applied repeatedly to couples of sectors and then (F_1, F_4) around V_1 produces V_1'' (see Figure 8b middle where F_1 and F_4 have been merged into F_j') and carry on distinguishing features around V_1 if possible after the edge merging process has been applied (see Figure 8b bottom). The resulting influence on the symmetry analysis is depicted in Figure 8c obtained if no splitting process is applied and Figure 8d when the splitting process has been applied that shows how the axisymmetry property is better delimited.

Another configuration depicted on Figure 8e exemplifies a configuration where F_3 and F_5 belong to the same surface S. Here, V_1 cannot be split because F_3 and F_5 depend from each other and F_2 cannot separate them.

This dependency reflects also configurations where surfaces are tangent to each other, e.g. if (F_1, F_2) are replaced by a C^1 surface that is tangent to (F_3, F_5), and V_1 coincides with a common point along the intersection curves between these surfaces.

In any case, it has to be pointed out that the vertex split operation does not modify the shape of the object and does not create new points over ∂M.

As a result of the above requirements, the corresponding transformations produce M_{MAX}, hence ∂M_{MAX}, which stands for the reference representation input to the divide phase. Requirements a and b can be formally expressed using the concept of maximal faces and edges while requirement c highlights the need for a classification of vertices prior to its formal statement. Also, the comparison of two geometric entities is subjected to tolerances, grouped under the designation ϵ, representing the tolerances used in a modeling kernel to state the coincidence of these entities. These tolerances express positional and angular differences between the compared entities.

Definition: Two faces F and F' of ∂M are homologous and belong to the same maximal face of ∂M_{MAX} if:

- they are generated from the same surface S, i.e. their reference surface is of same type, same intrinsic parameters and same location within ϵ;
- their orientation, as defined in the B-Rep datastructure, is identical;
- they are adjacent to each other through either one edge or one vertex, at least.

Consequently, maximal faces are generated through a face merging operator repeatedly applied to homologous faces adjacent along an edge.

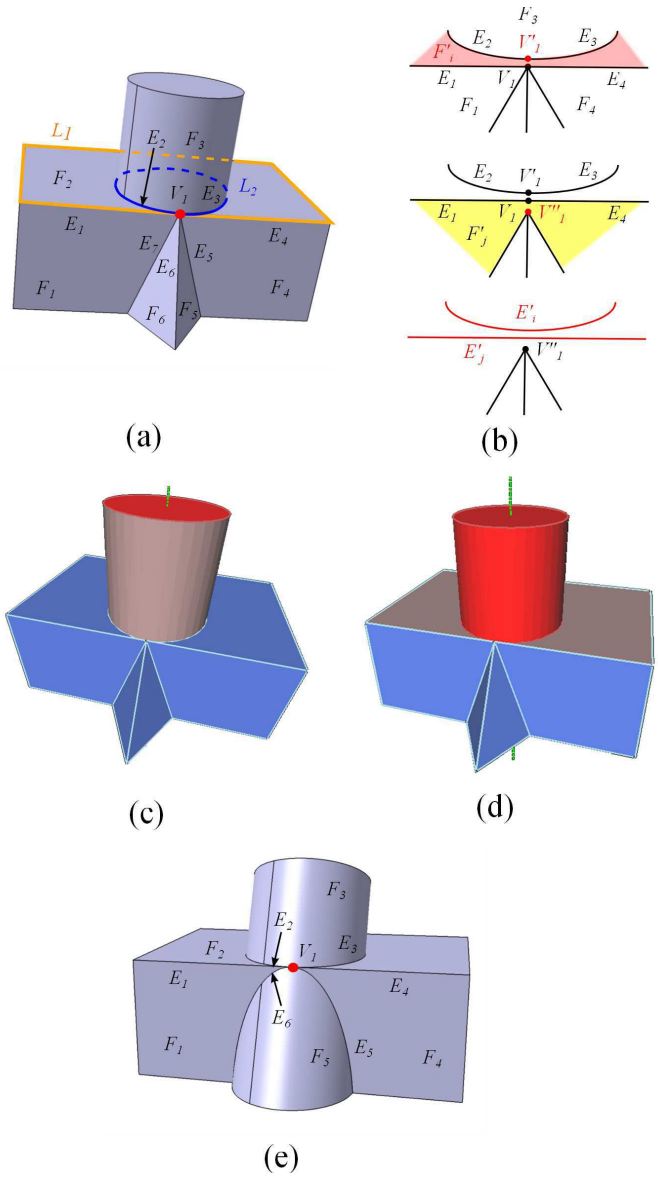


Figure 8: Configurations where: (a) V_1 can be split to characterize the independence of F_3 with respect to other faces, (b) effect of the vertex split at V_1 so that (E_2, E_3) and (E_1, E_4) are candidate for merging, (c) propagation of the axisymmetry property if V_1 is not split (color meaning is identical to Figure 6) , (d) propagation of the axisymmetry property if V_1 is split, (e) V_1 cannot be split because F_3 and F_5 depend from each other though F_3 is surrounded by F_2.

Definition: Two edges E and E' of ∂M are homologous and belong to same maximal edge of ∂M_{MAX} if they are the only edges connected to their common vertex V. In this case, E and E' are both adjacent to faces F and F' of ∂M_{MAX}. Hence, these edges define the same intersection curve.

Maximal edges are generated through an edge merging operator repeatedly applied at vertices connected to two edges only, after maximal faces have been generated.

Now, regarding the extension of faces through vertices, it is the purpose of the vertex split operator introduced at configuration [c] to handle this transformation when independent face subsets exist around a vertex. To characterize this independence, the concept of crossing configuration is set up (see Figure 9). A dual graph, restricted to the neighborhood of a candidate vertex V, is generated using the ordering of faces and edges around V, obtained from ∂M. The neighborhood of V being homeomorphic to a topological disk, a dual graph describing all the sectors of faces reduces to a simple loop and these sectors can be numbered counterclockwise around V as Φ_i, $i \in \{1, \ldots, n\}$. Around V, sectors of homologous faces are grouped together to form subsets (see Figure 9a and b), e.g. let us consider that there exists at least a couple of homologous sectors (Φ_i, Φ_j), $(i, j \leq n)$, $i < j$. If there exists at least another couple of homologous sectors: (Φ_k, Φ_q), $(k, q \leq n)$, (Φ_i, Φ_j) and (Φ_k, Φ_q) are said to cross each other if $i < k < j$ and $j < q \leq n$ or $1 \leq q < i$. When Φ_i is replaced by the identifier of the faces, faces having p sectors connected to V appears p times in the dual graph (see Figure 9). Figure 9b gives an example of crossing configuration.

A vertex with homologous faces in a non crossing configuration is a regular

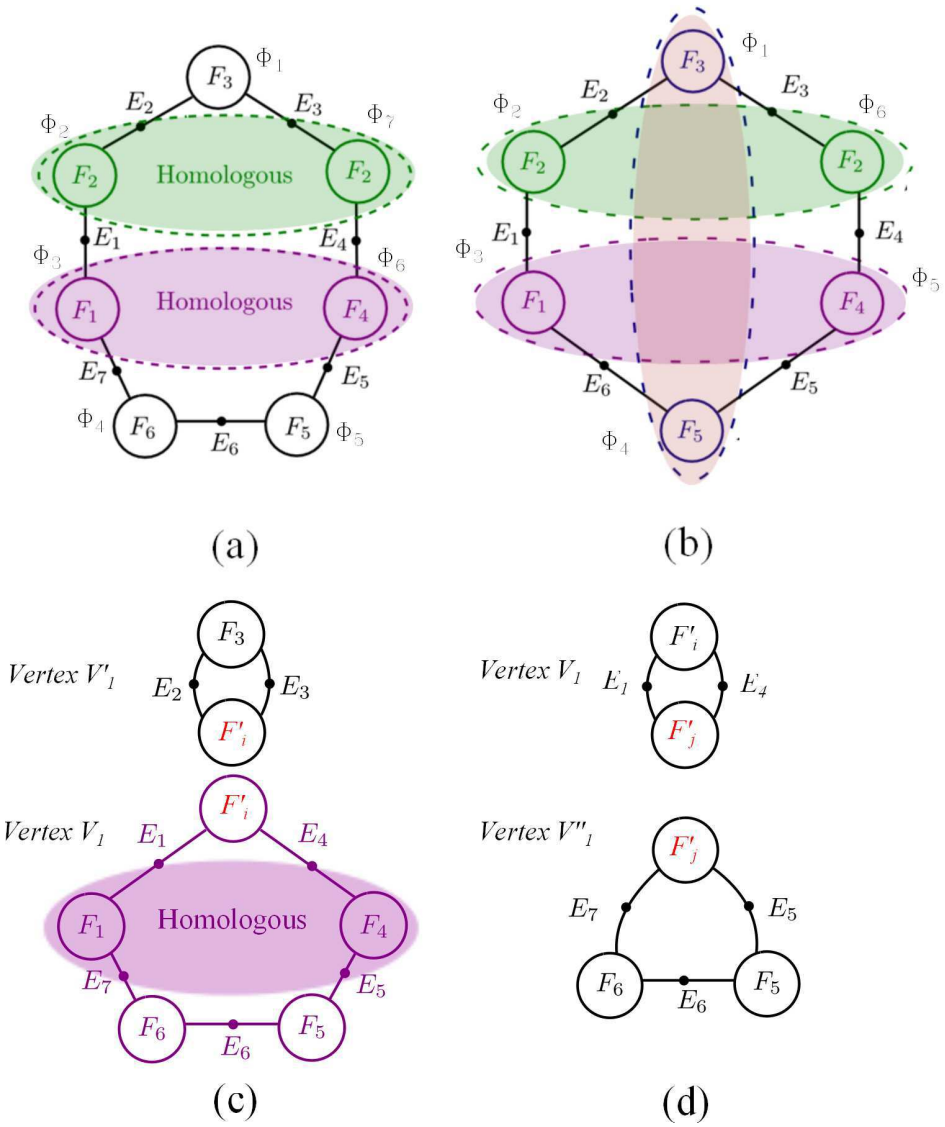


Figure 9: Dual graphs restricted to vertex V_1. Labels F_i as nodes of the dual graph indicates the relationship between faces of ∂M_{MAX} and sectors Φ_j: (a) dual graph illustrating a non-crossing configuration of Figure 8a, (b) dual graph illustrating the crossing configuration of Figure 8e, (c) dual graphs around V_1 and V_1' obtained after splitting V_1 once, (d) dual graphs around V_1 and V_1'' obtained after splitting V_1 a second time.

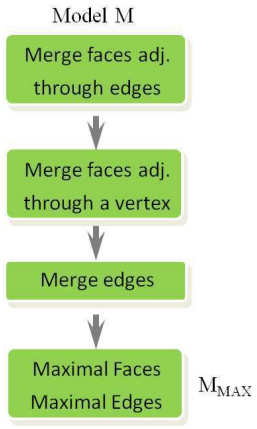


Figure 10: Sequence of operators transforming the boundary decomposition of M into M_{MAX}.

vertex and the vertex splitting operator can be applied using the homologous faces as separator of a face set (see Figure 9a, c). After a vertex split, the initial dual graph is split into two dual graphs reflecting the merging process between the homologous sectors (see Figure 9c when the vertex split has been applied at V_1 once and Figures 9d and 8b when it has been applied a second time). A vertex surrounded by homologous faces forming at least one crossing configuration and any number of crossing configurations that cannot be removed, is a non regular vertex. To process any arbitrary configuration, the vertex split is applied repeatedly to non crossing configurations that are independent of crossing configurations until each vertex is either a non regular vertex or a vertex with no homologous faces. The detailed algorithm to process arbitrary configurations is not described for sake of conciseness but its principle is a recursive application of the vertex split operator whenever there exists regular vertices.

Now, regarding the sequence of boundary decomposition transformations

required to obtain M_{MAX} from M, categories of operations can be chronologically ordered as follows (see Figure 10):

- face merging for homologous faces adjacent through edges;
- vertex split for regular vertices to merge faces adjacent through a vertex and produce maximal faces;
- edge merging for homologous edges to produce maximal edges.

As a result, ∂M_{MAX} contains edges where their adjacent faces are no more than G^1 continuous.

Taking into account the requirements to describe the boundary decomposition of M_{MAX}, the datastructure supporting the boundary decomposition transformations is currently of type hypergraphs to fit into a generic framework and complement the boundary decomposition needed in other product description like mesh generation constraints [37] rather than redefining a new and specific datastructure. Three hypergraphs are set up, G_{21}, G_{20}, G_{10}, where indices 0, 1, 2 designate vertices, maximal edges and faces, respectively. In the present case of volume objects G_{21} reduces to a graph because each maximal edge is exactly adjacent to two maximal faces, as long as ∂M_{MAX} contains one edge, at least. Section 7 gives an example of ∂M_{MAX} description.

5. Division phase of the symmetry analysis

The purpose of this section is to show that the symmetry analysis proposed can be achieved solely from the face type, location, orientation, intrinsic parameters and vertices coordinates of ∂M_{MAX}, i.e. without referring

to curves bounding faces and other key points that differ from the vertices of ∂M_{MAX}. Though a concise proof of the corresponding algorithm is not available, the strong connexion between the generation of ∂M_{MAX} and the division phase is emphasized with its geometrical and topological aspects:

- the division with respect to topological entities of ∂M_{MAX} that describes all the possible statuses of a point $P \in \partial M_{MAX}$ with respect to faces, edges and vertices of ∂M_{MAX} that can initiate a symmetry plane;
- the division with respect to geometric entities of ∂M_{MAX} where the geometry of each face is taken into account to characterize the existence of a CSP at $P \in \partial M_{MAX}$.

5.1. Division with respect to topological entities of ∂M_{MAX}

First of all, considering that any GSP cuts ∂M along loops according to the genus of M and M is 2-manifold, it can be observed that any of these loops cannot degenerate into a point $P \in \partial M$ or an open curve $C \in \partial M$. A GSP, Π, separates two neighborhoods and if it contains locally P only, M being 2-manifold, the neighborhood of P is topologically equivalent to a disk and all the points of this disk must lie outside Π, except P. Consequently, the neighborhood of P must lie entirely on one side of Π but Π being a GSP, this implies that there must exist another neighborhood of P, symmetrically located on the other side of Π and this neighborhood forms also a disk, which contradicts the fact that M is 2-manifold. Therefore, loops cannot degenerate to a point. A similar reasoning process holds for C, showing that loops obtained by $\Pi \cap M = \Pi \cap M_{MAX}$ is a set of loops, each of which being

a closed curve C_C lying on ∂M_{MAX} where every point has a topological disk cut by Π.

Now, let $P \in C_C$, P is located either inside F, a maximal face of ∂M_{MAX}, or at the boundary of F where P is shared by two faces, i.e. P belongs to a maximal edge E. E is shared by F and its adjacent face F'. According to the previous section, E can be either open and bounded by two vertices or closed and containing one or no vertex.

If E is open and P is located at the extremity of E, P coincides with V, a vertex of ∂M_{MAX}. V is shared by three faces at least according to the definition of maximal edges (see section 4).

If E is closed, it can contain one vertex only and, in this case V is a non regular vertex as defined in the previous section. If E defines a closed intersection curve bounding face F and F is adjacent to the face F', it is represented by a single edge and, according to the definition of maximal edges, E does not contain any vertex. If E is adjacent to other loops (see Figure 8a) involving other faces defined by surfaces differing from the surfaces defining both F and F', then let us assume that F' defines two sectors and only two around V. These two sectors being defined from F', they are homologous and the other faces adjacent at V being different from F and F', these two sectors cannot be crossed by any of the other faces around V. Therefore, V is a regular vertex that can be split along the two sectors defined from F' to merge them. As a result, the vertex lying on E after having split V is removed by the edge merging operator because, on both sides of V, E separates only F and F'. Consequently, if V is the only vertex on E after the generation of ∂M_{MAX}, V is a non regular vertex and at least one other sector

exists around V that contains a face homologous to F and defines a crossing configuration (see Figure 8e). Because V defines a crossing configuration, it is surrounded by four sectors at least, two of which at least related to F and two others at least are related to F'.

If E is closed and does not contain any vertex, it is a closed intersection curve between F and F'.

If P belongs to E and does not coincide with V, the neighborhood of P is defined by F and F' only.

As a summary of the above analysis, the reference configurations of any $P \in C_c$ enumerate:

- C_C contains points P that all lie inside a face of ∂M_{MAX}. This leads to a Π dividing a face (see Section 5.2.1). Because faces of ∂M_{MAX} can be bounded by an arbitrary number of loops, the number of loops influences the division (see Section 5.2.5);

- C_C contains points P that all lie on an edge since an edge can have no vertex. In this case, this edge divides exactly two faces and this is addressed at Section 5.2.3;

- C_C contains a point P that lies exactly on a vertex of an edge $E \in \partial M_{MAX}$ that connects (F_1, F_2). If E is open, it is bounded by two vertices and P coincides with one of them. Then, there exists another edge E' adjacent to E at P, that connects (F_2, F_3), to define a loop bounding the face F_2 of ∂M_{MAX}. E and E' identify three distinct faces (F_1, F_2, F_3) used to set Π, when it exists (see Section 5.2.4). If E is closed, P is indeed a non regular vertex and Π may divide E on each side of P and

E connects (F_1, F_2). There, only two faces are involved but considering the sectors around P that are separated by each side of E, the existence of Π reduces to the study of the triplet (F_1, F_2, F_1), which is covered by Section 5.2.4;

- C_C contains a point P that lies on an edge E of ∂M_{MAX} but does not coincide with a vertex of E if E has one. This leads to Π dividing an edge (see Section 5.2.2).

All other configurations of C_C are just a combination of these elementary ones. Hence, all the configurations of the division phase are covered.

5.2. Geometric division of ∂M_{MAX} for CSPs and CSAs

The above analysis defines all the possible locations of $P \in C_C$ with respect to ∂M_{MAX} and Π. Now, let us address the geometric neighborhood of P. Let P be a regular point of the surface defining face F. P lies in the interior of F and $P \in \Pi$ (see Figure 11). Because $P \in \Pi$, the tangent plane at P is orthogonal to Π, i.e. Π is a normal plane at P. P being regular, two distinct principal curvatures exist at P if P is not umbilical. Among the normal planes at P, Π can be aligned with either of principal directions of curvature D_1 or D_2 only, because the curvature distribution on both sides of a normal plane is symmetric only if this plane coincides with either of the principal curvature directions. If P is umbilical, Π can take any direction around the normal at P. Π being aligned with a principal curvature at $P \in C_C$ and assuming that C_C contains a regular arc around P, every point of this arc has the same property as P with respect to Π, therefore this arc is a line of curvature in F and this line is planar because it lies in

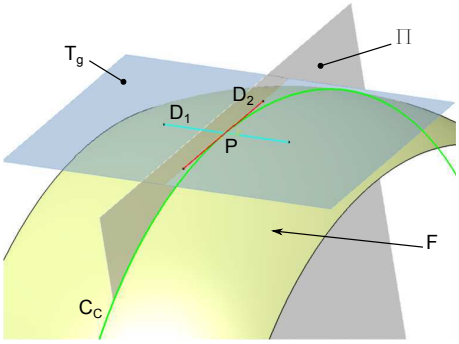


Figure 11: Local parameters at P to define a local symmetry plane Π.

Π. Given the set of primitives surfaces describing ∂M_{MAX} and considering their entire surface, valid lines for these surfaces can be any straight line on a plane, any generatrix or circle on a cylinder, any generatrix on a cone, any circle containing the center for a sphere, any small circle or the inner or outer circles on a torus.

If M is not restricted to a sphere or a torus, its GSPs contain some of these curves defined through the boundaries of the faces of ∂M_{MAX}. Then, each face of ∂M_{MAX} is bounded by edges, possibly bounded themselves by vertices. Locally, an edge is defined by its two adjacent faces and a vertex by the intersection of two edges. Using neighboring faces of each face F of ∂M_{MAX}, it is possible to define lines of curvature on F that are valid for every entity of the boundary of F and define the corresponding CSPs, some of these locally valid symmetry planes must coincide with GSPs, if any. Indeed, a GSP interacts with ∂M_{MAX} through a set of elementary configurations forming C_C as well as all the loops between this GSP and ∂M_{MAX}. It is the purpose now to identify all these elementary configurations that can define CSPs or CSAs and form a subset of C_C as well as any couple of loops involved

in the definition of a GSP or a GSA.

5.2.1. Dividing a face

If C_C is lying entirely inside F, i.e. C_C does not intersect any edge of ∂M_{MAX}, the location of Π solely depends on geometric properties of F if F is not bounded by edges of ∂M_{MAX}. This configuration applies only if F is defined by a closed surface and here, it is the case for spheres or tori among the set of primitives surfaces addressed. This is a specific trivial case where ∂M_{MAX} contains only one face, no edge and no vertex (see Figure 12a). In a general setting, F is bounded by one or more edges forming at least one loop and if C_C does not intersect any edge, it means that F is bounded by an even number of loops symmetrically located with respect to Π. Such configurations occur for cylinders or spheres or tori primitives only, given their basic symmetry properties when these loops are periodic. Indeed, for cylinders and tori, there are no more than two periodic loops whereas for a sphere, any couple of loops behaves similarly to periodic loops. Then, the purpose here is not to go into the details of the taxonomy of configurations, which can be easily derived. The loop symmetry property can be established from the relative position of faces adjacent to F along the edges forming its bounding periodic loops. In the simplest configuration where F is bounded by two such loops, L_1, L_2, each one containing a unique edge defined by the adjacent surfaces F_1 and F_2. The existence of Π for F is established if the surfaces defining F_1 and F_2 are symmetrically set with respect to Π (see Figure 12b). If Π exists, it is unique. Collectively, all these configurations are designated as Surface Symmetry CSP (SS-CSP) and produce a finite set of CSPs.

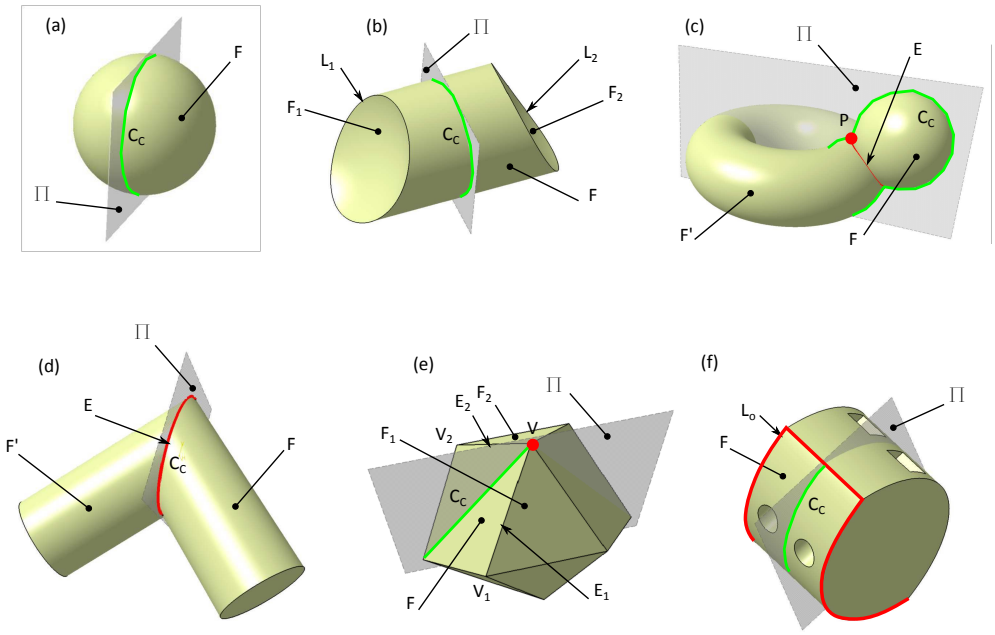


Figure 12: Taxonomy of configurations of symmetry planes Π intersecting M_{MAX} to define C_C. (a), (b) Π intersects a face F only that has no edge (a); separates loops L_1, L_2 (b). (c) Π intersects an edge E of F. (d) Π contains only one edge E of F representing the intersection between F and F'. (e) Π contains a vertex V and cuts F symmetrically with respect to F_1 and F_2 adjacent to F. (d) Π intersects F that is bounded by multiple loops. Π cuts L_o and separates the other loops.

5.2.2. Dividing an edge

If C_C partly lies inside F, C_C cuts at least one edge E bounding F at P. If P is not a vertex of ∂M_{MAX}, its neighborhood is defined by two faces F and F'. If E is at least G^1 at P, the tangent at P to E must be orthogonal to Π in addition to being located on planar lines of curvature in F and F' both (see Figure 12c). If E is G^0 at P, then the half tangent at P to E must be symmetrically located with respect to Π in addition to being located on planar lines of curvature in F and F' both. Given the definition of maximal edges, the latter configuration may occur when the intersection curve contains singular points.

The above configurations may not always define a finite set of points where CSPs can be assigned, even though E does not define an axisymmetric configuration, e.g. if F and F' are planar, any point P lying on their common edge E can be the location of a CSP Π orthogonal to E. To this end, if E is open, the locations of its two extreme vertices (V_1, V_2) in ∂M_{MAX} being independent of any parameterization, these location can be used to uniquely select Π. If E is closed, it means that E is a line of curvature common to F and F' and, in the present set of primitive surfaces, E is necessarily a circle, hence the local configuration is axisymmetric and processed as such with a CSA. It is worth pointing out that if E is open, the location of (V_1, V_2) can be used to filter out CSPs where these vertices are not symmetrically located with respect to Π.

All the corresponding configurations can now produce a finite set of CSPs, designated as Orthogonal CSP (O-CSP), uniquely defined and forming a taxonomy that can be derived from the various combinations and locations

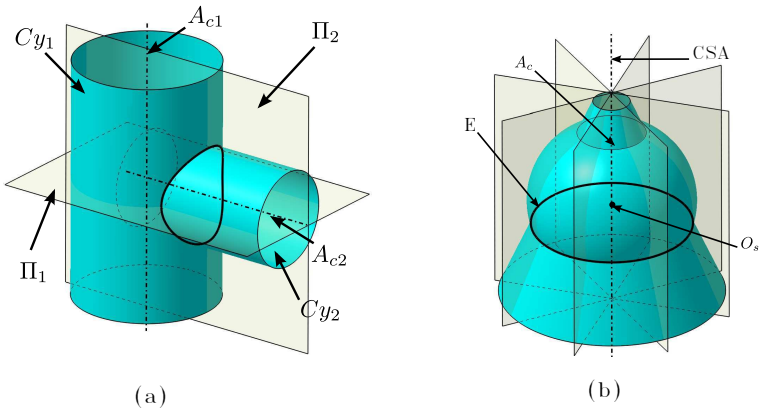


Figure 13: (a) An example taken from the taxonomy with two orthogonal cylinders producing two O-CSPs Π_1 and Π_2. (b) Another example from the taxonomy with a cone and a sphere producing a CSA.

of couples of primitives (see Figure 13). These configurations produce also the CSAs, uniquely defined through a taxonomy of couples of primitives.

5.2.3. Dividing two faces

If C_C lies on the boundary of F, it contains one and only one edge E of F. E must be a planar intersection curve between F and its adjacent face F' (see Figure 12d). Then, the existence of a CSP Π is validated if F and F' are of same type, same intrinsic parameters and if they are symmetrically located with respect to Π. If C_C contains two adjacent edges bounding F, these edges are connected to each other through a vertex V of ∂M_{MAX}. V is connected to three different faces at least (see section 5.1). Therefore, if Π contains two successive edges of F, the second one is adjacent either to a face differing from F', hence this edge cannot belong to Π, or to a face homologous to F' but this means that V is a regular vertex that can be split to merge the two homologous sectors related to F', which contradicts the fact that ∂M_{MAX} cannot contain regular vertices. All the corresponding configurations can

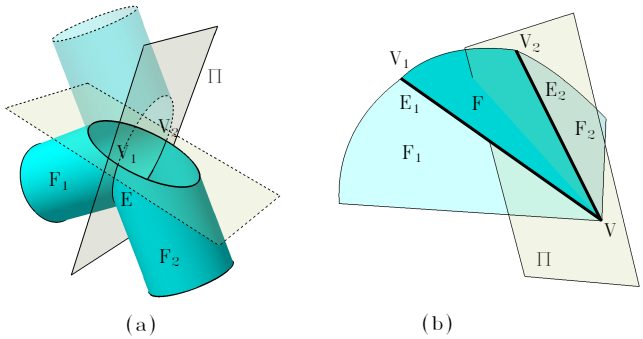


Figure 14: (a) An example from the taxonomy with two cylinders defining a BS-CSP configuration at E. (b) An example from the taxonomy with F bounded by two cylinders F_1 and F_2 that define an LB-CSP at V.

now produce a unique CSP per edge designated as Bisector CSP (BS-CSP) forming a taxonomy that can be derived from the various combinations and locations of couples of primitives (see Figure 14a).

5.2.4. Dividing two edges

If C_C contains only a vertex V of F, V is shared by two other faces at least (see Section 5.1). Around V, F can be chosen as reference face to determine whether or not a line of curvature can define a CSP Π that contains V only. At V, if F is bounded by two edges (E_1, E_2) defining the adjacency of F with F_1 and F_2, these edges are open and have each an extreme vertex opposite to V (see Figure 12e). There can be either one, two or a infinite number of curvature lines defining CSPs of F passing through V whether V is umbilical or not. To select a valid one if any, let us consider V and its two adjacent edges. A valid CSP containing V and lying on F must separate F_1 and F_2. If Π can be generated that contains V and is a bisector plane for F_1 and F_2 and Π intersects F along a curvature line, then Π is a valid CSP for F and its adjacent faces at V. If it exists, the CSP Π is uniquely defined and,

taking into account the location of extreme vertices (V_1, V_2) of (E_1, E_2), if V_1 and V_2 are symmetrically located wrt the CSP, its validity is extended with the effective bounds of (E_1, E_2) in F. If F is bounded by one edge, V is a non regular vertex and the validity of the CSP extends to the whole edge. All the corresponding configurations can now produce a unique CSP at a vertex for any triplet of adjacent faces and it is designated as Loop Bisector CSP (LB-CSP) forming a taxonomy that can be derived from the various combinations and locations of couples of primitives (see Figure 14b).

5.2.5. Dividing a face with multiple loops

Finally, it is mandatory to consider that faces of a B-REP CAD solid can be bounded by an arbitrary number n of loops and how transitions between couples of loops can be achieved. Therefore, if C_C lies on such a face F, the corresponding CSP Π can either separates all the components of ∂F (see Π_1 Figure 15a) or cut at least two components (see Π_2 Figure 15a) or cut one component and separates the others (see Figure 15b) depending on n being odd or even. The first case is indeed the configuration defining SS-CSPs. The second one means that the CSP defines a part of C_C that is initiated by an O-CSP or an LB-CSP on one loop and coincides with a CSP of either of these types on another loop. Therefore, this configuration is covered by the previous ones. The latter configuration differs from the others because C_C cuts one loop, L_0, and separates others (see Figure 12f). The corresponding CSP is called Loop Symmetry CSP (LS-CSP). If L_0 has been assigned a finite number of CSPs, the validity of Π for F is checked when the other loops are symmetrically set with respect to Π. If L_0 has a CSA assigned, this one is used and F is either a plane bounded by L_0 or it is a

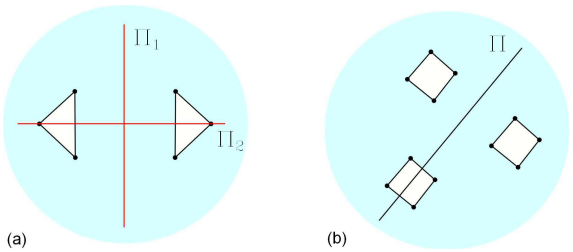


Figure 15: (a) A face F bounded by an even number of loops with Π_1 (SS-CSP) and Π_2 (defined from O-CSPs or BS-CSPs). (b) A face F bounded by an odd number of loops with Π (LS-CSP).

surface of revolution. Then, a grid of features characterizing the loops with regard to the CSA is built to speed up the search for symmetric loops. The barycenter of loop vertices or the center point of loop edge derived from the axis of its reference surface defines a key point, P_k, attached to each loop. Each point of this grid is defined by three parameters that are necessarily coincident: the first one is the distance from P_k to the CSA, the second is the location of P_k along the CSA and the third one indicates the number of maximal edges in the loop. Now, considering a candidate loop L_1, with its corresponding key point P_k, a geometric search looks for a coinciding grid point P'_k having the same first and third components and the corresponding loop L_2 is identified. Then, an LS-CSP is generated after loops L_1 and L_2 are effectively checked for symmetry since key points are not sufficient to compare loops. When considering cylinders, other LS-CSP can be generated using a geometric search based on the first and third key point coordinates. Cones, tori and spheres are processed with grids adapted to their features.

Because all the configurations of interactions between a GSP or a GSA and ∂M_{MAX} have been defined and any arbitrary loop C_C can be generated from them, all possible configurations of GSPs or GSAs with ∂M_{MAX} are

available after the division phase. All the CSPs and CSAs are obtained independently of face and edge parameterizations and the only tolerance used is ϵ to stay consistent with modeling kernels. Algorithm 1 synthesizes the division process where $T[]$ is the list containing all the CSPs and CSAs and the corresponding faces where they are attached. Now that the division phase is able to assign the LB-CSPs, O-CSPs, BS-CSPs, LS-CSPs, SS-CSPs and CSAs to all the appropriate entities of ∂M_{MAX} to form any loop C_C and any number of loops, the conquer phase can be initiated to extend these local symmetry properties up to a global level for some of them.

Function `Division(`∂M_{Max} *data*`)/* generates CSPs & CSAs */`
Data: Maximal faces, maximal edges, maximal loops of ∂M_{Max};
Result: $T[1, ..., n]$: CSPs and CSAs attached to ∂M_{Max} entities;
$n \leftarrow 0$;
for *each edge* E_i **do**
  $(F_1, F_2) \leftarrow$ faces adjacent to E_i;
  $T[n++] \leftarrow$ O-CSP(E_i, (F_1, F_2)) generated from O-CSP taxonomy;
  if *CSA taxonomy generates a CSA from* E_i, (F_1, F_2) **then**
    $T[n++] \leftarrow$ CSA(E_i, (F_1, F_2));
for *each edge* E_i **do**
  $(F_1, F_2) \leftarrow$ faces adjacent to E_i;
  $T[n++] \leftarrow$ BS-CSP(E_i, (F_1, F_2)) generated from BS-CSP taxonomy;
for *each vertex* V_i **do**
  for *each loop* L_j *containing* V_i **do**
    $(E_1, E_2) \leftarrow$ edges $\in L_j$ sharing common vertex V_i;
    $T[n++] \leftarrow$ LB-CSP($L_j, (E_1, E_2), V_i$) generated from LB-CSP taxonomy;
for *each face* F_i **do**
  if F_i *is a torus* ***and*** *has no edge* **then**
    $T[n++] \leftarrow$ CSA generated on F_i axis;
  if *(*(F_i *is a torus* ***or*** F_i *is a cylinder* ***or*** F_i *is a sphere)* ***and*** *has periodic loops* (L_1, L_2)*)* **then**
    $(F_1, F_2) \leftarrow$ faces adjacent to F_i;
    $T[n++] \leftarrow$ SS-CSP($(L_1, L_2), F_i$) generated from SS-CSP taxonomy;

Algorithm 1: Division process for CSPs and CSAs generation

6. Local and global symmetry properties through propagation processes

The conquer phase aims at creating GSPs and a possible GSA if the object is effectively axisymmetric. In accordance to section 3.2, the symmetry

analysis is performed through two successive propagation processes. Each of them relies on a front propagation based on the coincidence of CSPs or of CSAs or both. The first propagation process generates sets of faces that define loops, or part of loops, C_C between ∂M_{MAX} and the extent of symmetry planes or symmetry axes. During the first propagation process, the coincidence of CSPs is satisfied when, for a given face F and either an O-CSP or an LB-CSP separating ∂F, this CSP can be propagated to adjacent edges of ∂F if the faces F_1 and F_2 adjacent to F on each side of the CSP are symmetrically located with respect to it. Then, the second propagation process aims at extending the set of faces in ∂M_{MAX} that are symmetric wrt a CSP to cover the largest possible area of ∂M_{MAX}. During the second propagation process, for two given faces F_i and F_j located on each side of a CSP, Π, and one edge on each of these faces E_i and E_j (see Figure 17), Π can be propagated to adjacent edges of ∂F_i and ∂F_j if the faces F_i' and F_j' adjacent to F_i and F_j on each side of Π are symmetrically located with respect to it. These propagation conditions share the following common concepts.

Concave and convex face configurations must be distinguished. Here, surface orientation available from the B-Rep description of M is combined with the curvature distribution of each surface to be able to distinguish convex and concave configurations (see Figure 16). This leads to a quantity called the orientation index, O_i, that is point independent and, consequently, parameterization independent. Considering the unit vector $\mathbf{n}_f$ at a point $P \in F$ of ∂M_{MAX}. To reflect the curvature distribution of F, $\mathbf{n}_f$ is colinear to the normal $\mathbf{n}$ at $P \in F$ and its orientation is such that it points toward a center of curvature. The center of curvature is a point:

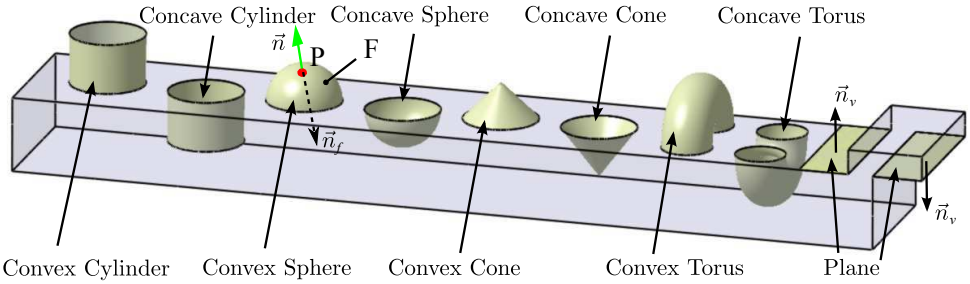


Figure 16: Configurations showing the effect of O_i to distinguish concave and convex areas.

- on the axis for a cylinder and a cone;
- coinciding with the center for a sphere;
- on the circle of average radius for a torus;

using geometric properties of one of the principal curvatures of these surfaces. In case of a planar face, $\mathbf{n}_f$ is not defined because all curvatures vanish resulting in no center of curvature at finite distance. As a complement, the orientation of ∂M_{MAX} defines the orientation of each face F, with the corresponding normal orientation $\mathbf{n}_v$, $\forall P \in F$. Then, $O_i = \mathbf{n}_f.\mathbf{n}_v = \pm 1$ distinguishes convex and concave configurations of F without referring to a specific point and is insensitive to tolerances. Figure 16 illustrates configurations where the O_i is mandatory to correctly analyze symmetries. If F is planar, the comparison of faces reduces to that of their normal $\mathbf{n}_v$, which is sufficient to indicate the material side. Consequently, F_i' and F_j' must have the same orientation index to be symmetric with respect to Π. This condition is strictly necessary when E_i and E_j have no vertex.

In addition to the orientation index, symmetric faces F_1 and F_1' bounded by multiple loops described by a single edge without a vertex produce ambigu-

ous configurations with respect to a given CSP Π when these loops connect F_1 and F_1' to F_2 and F_2', respectively (see Figure 17). If the intersection between the surface S_1 defining F_1 and S_2 defining F_2 produces multiple intersection curves that can bound M differently while M is still a solid, complementary tests are needed to check that the faces used in M are effectively symmetric wrt Π. This configuration occurs when F_2 is not connected to any other face, as in Figure 17 and a taxonomy of configurations is derived from the previous conditions that involves (cylinder, torus), (cone, torus), (torus, torus). Anyhow, the principle used discriminates loop locations representing different intersection solutions between two surfaces, F_1 and F_2 using an arbitrary point P set at the boundary of a face F_1 where loop locations are ambiguous, $P \in \partial F_1$. F_1 is defined by the surface $S_1(u_1, v_1)$ and P is located as $P = S_1(u_{1P}, v_{1P})$. Because F_1 and F_1' are of same type and already symmetric with respect to Π, they are represented with the same nature of parameterization but possibly have different parameterization origins. Then, P is transformed into P' using Π, i.e. $P' = \Pi(P)$. To simplify the presentation, let us assume that parameterization origins are symmetrically set, i.e. F_1' is defined by $S_1'(u_1', v_1') = \Pi(S_1(u_1, v_1)) = S_1'(u_1, v_1)$ and $P' = S_1'(u_{1P}, v_{1P})$. Now, P' can be compared to P'', $P'' \in \partial F'$, a point defined on a loop of F_1' in its parameter space. If $\|P', P''\| < \epsilon$, the corresponding loops are effectively symmetric wrt Π. If parameterization origins are not symmetrically set, P' is defined by taking into account the offset between origins to avoid any influence of parameterization in locating P''. If the loops being compared contain singular points, i.e. $\exists Q \in F_1$, $T_p(Q) \perp \Pi$, where $T_p(Q)$ designates the tangent plane at Q to F_1 and Q is a point on a loop, these points must

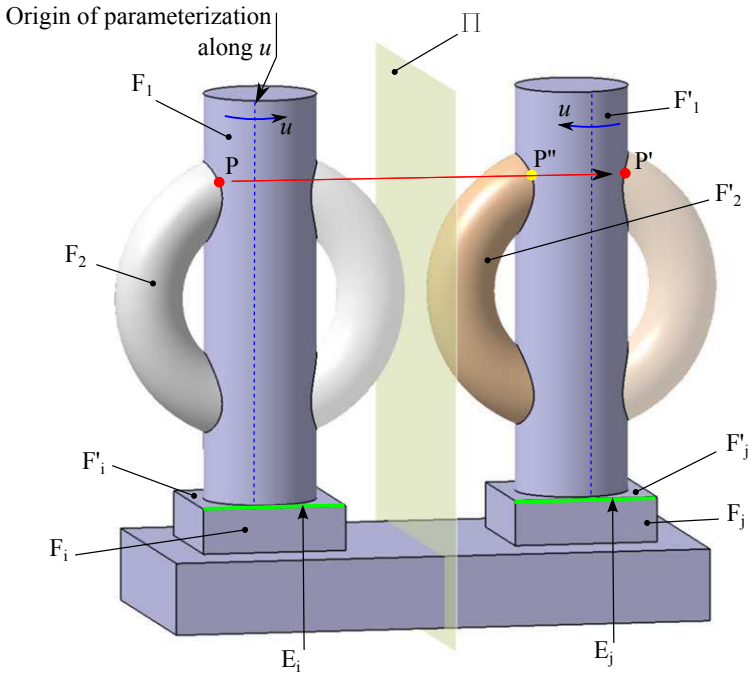


Figure 17: Front propagation from $((F_i,E_i), (F_j,E_j))$ to F'_i, F'_j and comparison of loops to remove ambiguous configurations (the toroidal faces F_2 and F'_2 have alternate solutions (visualized as transparent faces) that must be analyzed using a reference point $P \in F_1$ mapped to the candidate face F'_1. Here, P' differs from P'' showing asymmetry between F_1 and F'_1 loops).

be avoided because Q and $\Pi(Q)$ can belong to two different loops.

6.1. First level propagation process

The first level decomposes as follows (see Algorithm 2). $T[]$ denotes the list of CSPs, CSAs obtained from the division phase. Each $T[i]$ contains a list of faces satisfying the symmetry represented by its CSP or CSA. The first level has the purpose of creating the so-called 'CSP chains' starting from any O-CSP, LB-CSP or BS-CSP and propagating across adjacent faces and their corresponding loops, as long as the new CSPs encountered coincide with the initial one. 'CSP chain' is a common designation of a collection of faces

sharing coplanar CSPs, co-linear CSAs and CSAs lying in a given CSP. This propagation process stops either when the set of faces spanned forms a loop, representing the location of the intersection curve C_C between this CSP and ∂M_{MAX}, or when there is no more CSP coinciding with the initial one, i.e. showing the local asymmetry of ∂M_{MAX}. During this propagation, CSAs can alternate with CSPs to expand the axisymmetric areas and connect with CSPs if needed.

Then, processing the faces already symmetric wrt a given CSP Π that are bounded by multiple loops includes the update of their symmetry properties using LS-CSPs that do not cut any loop L_i and are attached to their corresponding face F. Indeed, during the overall propagation, if it reduces to the propagation of CSAs only, it produces a GSA. If so, a propagation process takes place to look for a possible GSP orthogonal to the GSA. All the merging processes involved in Algorithm 2 imply a decrease of $\texttt{Card}(T)$.

Finally, `Merge_Coinciding_CSP` looks for disconnected sets of faces that coincide and merge them to describe all the loops or part of loops that fall into the same CSP when M is cut by this CSP. It has to be noticed that Algorithm 2 computes GSPs/GSA as well as local symmetries simultaneously since this propagation process is not stopped if intersection curves C_C between CSPs and ∂M_{MAX} don't form loops.

```
Function Propagation_Level_1(T[1,..,n] list of CSPs and CSAs, ϵ)
        Result: CSPs and CSAs forming so-called CSP chains;
while T[i] ∈ T[] not visited do
    for each T[i] ∈ T[] that is CSP typed do
        if T[i] is O-CSP typed then
            E_R ← self-symmetric edge attached to T[i];
            (F_1, F_2) ← symmetric face pair adjacent to E_R;
            Propagate_O-CSP(T[i], F_1, E_R, ϵ);
            Propagate_O-CSP(T[i], F_2, E_R, ϵ);
        if T[i] is LB-CSP typed then
            (E_1, E_2) ← symmetric edge pair attached to T[i];
            Propagate_LB-CSP(T[i], E_1, E_2, ϵ);
    Merge_All_Coinciding_CSP_CSA(T[], ϵ);
for each T[i] ∈ T[] do
    if Face= T[i] is Mult-Loop typed then
        Update_SymProp_LS-CSP(T[], ϵ);
if T[1] is CSA typed and Card(T)= 1 then
    Propagate_SS-CSP(Card(T, ϵ));
    exit function;
else
    while T[i] ∈ T[] not visited do
        for each T[Card(T) −j] ∈ T[] that is CSP typed do
            if T[i] coincide T[Card(T) −j] ±ϵ then
                Merge_Coinciding_CSP(T[i],T[Card(T) −j]);
```

Algorithm 2: First level propagation process

6.2. Second level propagation process

Now, the second level propagation process can start to identify the GSPs or determine the areas where local symmetry properties stop. Its principle expands the areas attached to each CSP chain on both of its sides wrt this CSP Π using a front propagation process for each side. The overall process is described in Algorithm 3. One condition to stop the propagation is when all the faces of ∂M_{MAX} have been visited. In this case, Π attached to these

fronts becomes a GSP. Otherwise, the propagation ends if there is no new candidate couple of faces available, which produces the area S_a of ∂M_{MAX} that is valid for this CSP. S_a characterizes local symmetry properties in ∂M_{MAX} and may contain several connected components. Both conditions are characterized in Algorithm 3 by an empty $front_list[]$. The condition to propagate both fronts is obtained if the new faces added to the fronts are symmetrically set wrt Π. To identify couples of candidate faces (F_i', F_j'), this propagation process uses edge adjacencies between the already identified symmetric area S_a using (F_i, F_j) that are faces part of the front and (E_i, E_j) that are edges bounding F_i and F_j, respectively, and that are contained in each front on each side of Π (see Figure 17). Using adjacency relationships, (F_i, F_j) and (E_i, E_j) uniquely define (F_i', F_j'), the faces adjacent to E_i and E_j, respectively. Then, the couples (F_i, F_i') and (F_j, F_j') can be used to check if they are symmetrically set wrt to Π. Then, the couples (F_i, F_i') and (F_j, F_j') can be used to check if they are symmetrically set wrt to the CSP. This is the reference configuration used to propagate the front similar to that described in a particular setting at the beginning of Section 6 (see Figure 17) but here the edges E_i and E_j are bounded by two vertices, which stands as the general configuration. This configuration is processed in `Propagate_Front_On_Loops` where all the loops of (F_i', F_j') are processed. Ambiguous configurations are also processed as special cases of loops within this procedure. During the propagation process, not only faces visited are marked but edges too, in order to process them once.

Function `Propagation_Level_2(`*T[1,..,n] list of CSPs, ϵ*`)` **for** *each* $T[i] \in T[]$ *that is CSP typed* **do**
    for *each* (E_1, E_2) *pair of symmetric edges attached to* $T[i]$ **do**
        $(F_1, F_2) \leftarrow$ symmetric faces adjacent to (E_1, E_2);
        add to $front_list[] \leftarrow ((F_1, F_2), (E_1, E_2), T[i])$;
repeat
    $((F_1, F_2), (E_1, E_2), T[i]) \leftarrow$ first element in $front_list[]$;
    `Propagate_Front`$(front_list, (F_1, F_2), (E_1, E_2), T[i], \epsilon)$;
    remove $((F_1, F_2), (E_1, E_2), T[i])$ from $front_list[]$;
until $front_list[]$ *is empty* ;
if *All faces* $F[j] \in \partial M_{MAX}$ *attached to* $T[i]$ *have been marked 'visited'* **then**
    $T[i]$ is marked as GSP;

Algorithm 3: Second level propagation process

Function `Propagate_Front(`$front_list, (F_1, F_2), (E_1, E_2), T, \epsilon$`)`
if F_1 *and* F_2 *do not have the same number of loops* **then**
    exit function;
$(L_1, L_2) \leftarrow$ loops of F_1, F_2 containing E_1, E_2 respectively;
`Propagate_Front_On_Loops`$(front_list, (F_1, F_2), (E_1, E_2), T[i],$
$(L_1, L_2), \epsilon)$;
remove from $front_list \leftarrow ((F_1, F_2), (E_1, E_2), T)$;

Algorithm 4: Front propagation for level 2

When the propagation process terminates and all faces of ∂M_{MAX} have been visited and checked for symmetry, it means that the CSP Π is indeed a GSP.

Finally, the symmetry analysis of M_{MAX}, hence of M, relies on the comparison of vertices and faces locations and faces parameters of ∂M_{MAX}. These comparisons can be performed within the tolerance ϵ of a CAD modeler so that the symmetry properties identified can be used in subsequent modeling operations to reconfigure a construction tree or other geometric transformations.

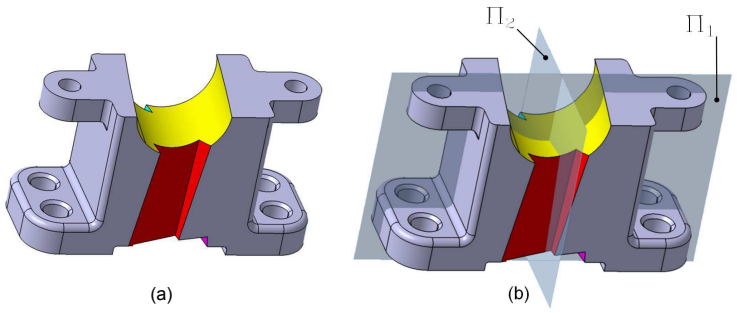


Figure 18: (a) A modified version of component 19 in the repository (see Figure 21). (b) No CSP in the modified area (yellow, red colors) can coincide with Π_2 to define a local symmetry in the gray areas.

Though the division and propagation processes can identify all the GSPs or GSA of M and perform simultaneously a local symmetry analysis, the local symmetries identified don't form the complete set of local symmetry planes because the division phase assumes that any CSP must define at least a subset of a C_C. As an illustration of this point, Figure 18a shows a modified version of component 19 in the model repository (see Figure 21) where pockets of directions differing from that of the gray areas, have been added all around the middle area of the solid, i.e. yellow, red, areas visible. Now, if Π_1 is a GSP for component 19, it is still intersecting the solid of Figure 18a along some valid curvature lines defining CSPs (see Figure 18b) and local symmetry covering the gray areas can be obtained. At the opposite, the other GSP for component 19, Π_2, is no longer intersecting the solid along valid curvature lines. As a result, there will be no CSP coinciding with Π_2 and the local symmetry covering the gray area wrt Π_2 cannot be identified. However, it can be observed that this lack of local symmetry could be solved with a new category of CSP but this configuration refers also to repetitions of primitives and has been left for future investigations.

7. Results and examples

The algorithm complexity and validity of results are demonstrated in this section. Although no formal proof of the overall algorithm is proposed in this article, the autors haven't observed any case where local or global symmetries are missed or are invalid.

7.1. Settings for result analyses

In a first place, the algorithmic complexity of the proposed approach can be evaluated as follows. Let us consider that the complexity of a B-Rep model is characterized by the numbers of vertices, n_V, edges, n_E, faces, n_F, of the input model M. The generation of ∂M_{MAX} from M (see Section 4) involves the face merging, vertex split and edge merging operations. If n_{V_F} is the maximum number of faces connected at a vertex, ∂M_{MAX} is obtained in $O(n_E)$, $O(n_V n_{V_F})$, $O(n_V)$ respectively for the previous operators. Because n_V, n_E, n_F are linearly related to each other by the Euler's theorem [34], $O(n_V n_{V_F})$ reduces to a quadratic complexity with respect to n_F: $O(n_F^2)$.

Then, the division phase uses M_{MAX} whose complexity is characterized by n_{V_M}, edges, n_{E_M}, faces, n_{F_M}. O-CSPs and BS-CSPs are obtained linearly in $O(n_{E_M})$. LB-CSPs are obtained in $O(n_{V_M} n_{V_F})$. Though the Euler's theorem is no longer applicable to M_{MAX}, it can be used as upper bound to relate V_M, E_M and F_M, hence the generation of LB-CSPs is $O(n_{F_M}^2)$. Let n_L be the maximum number of loops bounding a face, the LS-CSP generation is based on the geometric search whose complexity is $O(n_L \log n_L)$ in the worst case.

Finally, the propagation of first level can be performed in $O(n_{V_M} + n_{E_M})$ in the worst case for each CSP chain generation, as a first approxima-

tion. The complexity of the second level propagation can be estimated as $O(n_{F_M} n_{L_m} \log n_{L_m})$ for each CSP propagation where n_{L_m} is the maximum number of loops in a face, considering that the face complexity is dominated by the geometric search process. A more precise complexity analysis would incorporate the loop comparison but the corresponding increase of complexity would require more parameters like the maximum number of edges in a loop, which reduces the interest of this formal analysis and promotes an experimental approach that will be addressed later in this section.

To evaluate the efficiency and the level of performance of the algorithm, several examples are proposed. All the time measurements have been performed on a PC with an Intel Core i7-3740QM @ 2.70 GHz and 16GB memory. However, the largest model processed, among those mentioned in this section, did not used more than 1GB. Not only time measurements have been performed but a reference time evaluation based on pseudo clock cycles is proposed that is independent of the processor performance and can be used for comparison [41]. It is a hardware-based, small resolution cycle counter having a much smaller resolution than classical system time functions. The coherency between effective time measurements t (ms) and the amount of reference cycles c has been evaluated for all the examples and fits with a constant of $t = C_t\ c$ with $C_t = 3.71559\ 10^{-7}$. C_t fits within $\pm 0.005\%$ to the timings performed. Consequently, most of the performance analysis is based on time measurements but C_t provides the conversion to the amount of reference cycles for the processor used in our tests. The software development is based on OpenCascade library and the tolerance ϵ of position of reference surfaces has been set to $10^{-3} mm$, similar to the tolerance value

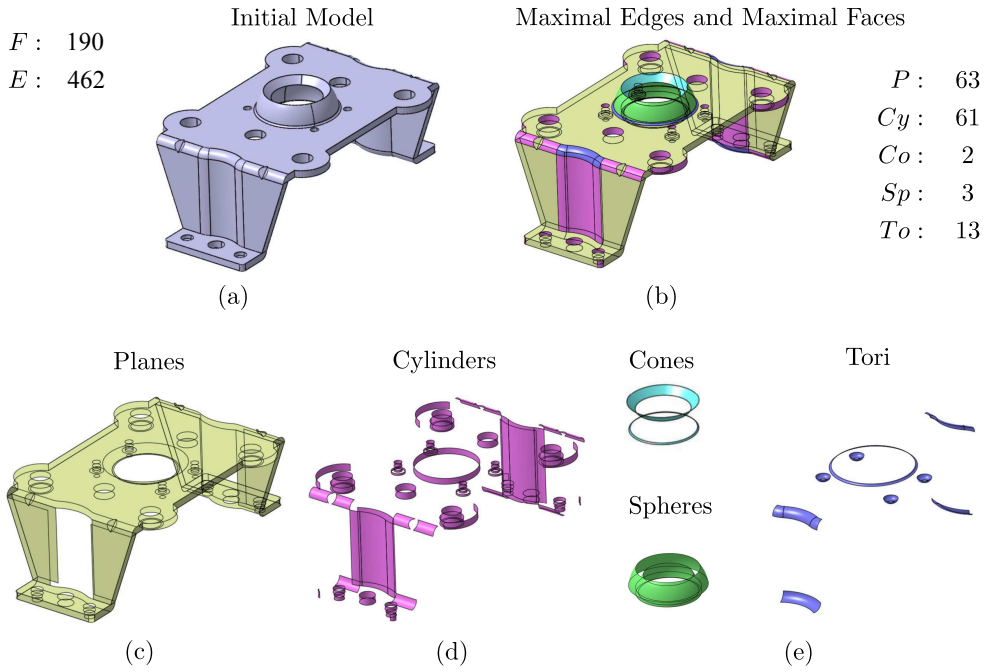


Figure 19: Example of the preparation phase. (a) E and F indicates the number of edges and faces of the initial model M. (b) P, Cy, Co, Sp, To state the numbers of surfaces per face type. (c, d, e) show the corresponding sets of surfaces.

used in CATIA V5. A detailed example is based on a mechanical component (see Figure 19) that illustrates the major steps of the symmetry analysis. The preparation process producing ∂M_{MAX} from M generates $F_M = 142$ faces from the 186 initial ones and $E_M = 273$ faces from the 446 initial ones.

Figure 20c illustrates the results of the propagation phase and the two GSPs found for this component. The total processing time is 25ms, which is very satisfactory to reach the interactive time needed for modeling applications. To enable the analysis of local symmetries, a graphic representation of the extent of a symmetric area with respect to a given CSP Π, is set up as follows:

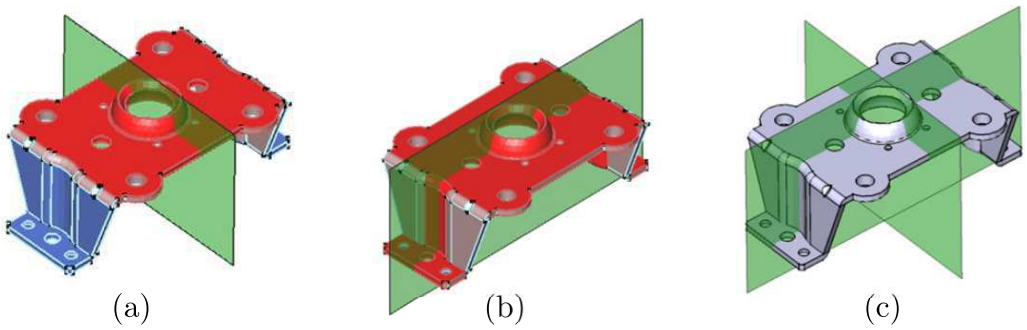


Figure 20: (a, b) Illustration of the first level propagation process applied to the component of Figure 19 with two CSPs. The colored areas follow the interpretation given in the text. (c) The two GSPs obtained for this component.

- vertices of ∂M_{MAX} are represented with small black squares;
- if a maximal face F as well as all its boundary edges are symmetric with respect to Π, F is colored in red;
- if a maximal face F is symmetric with respect to Π but at least one of its boundary edge is not, F is colored in light pink;
- if a maximal face F is not symmetric with respect to Π, it is colored in blue;
- maximal edges that are symmetric with respect to Π are colored in light red;
- maximal edges that are not symmetric with respect to Π are colored in light blue;
- an iconic representation of Π is represented with a transparent rectangle positioned with respect to the component.

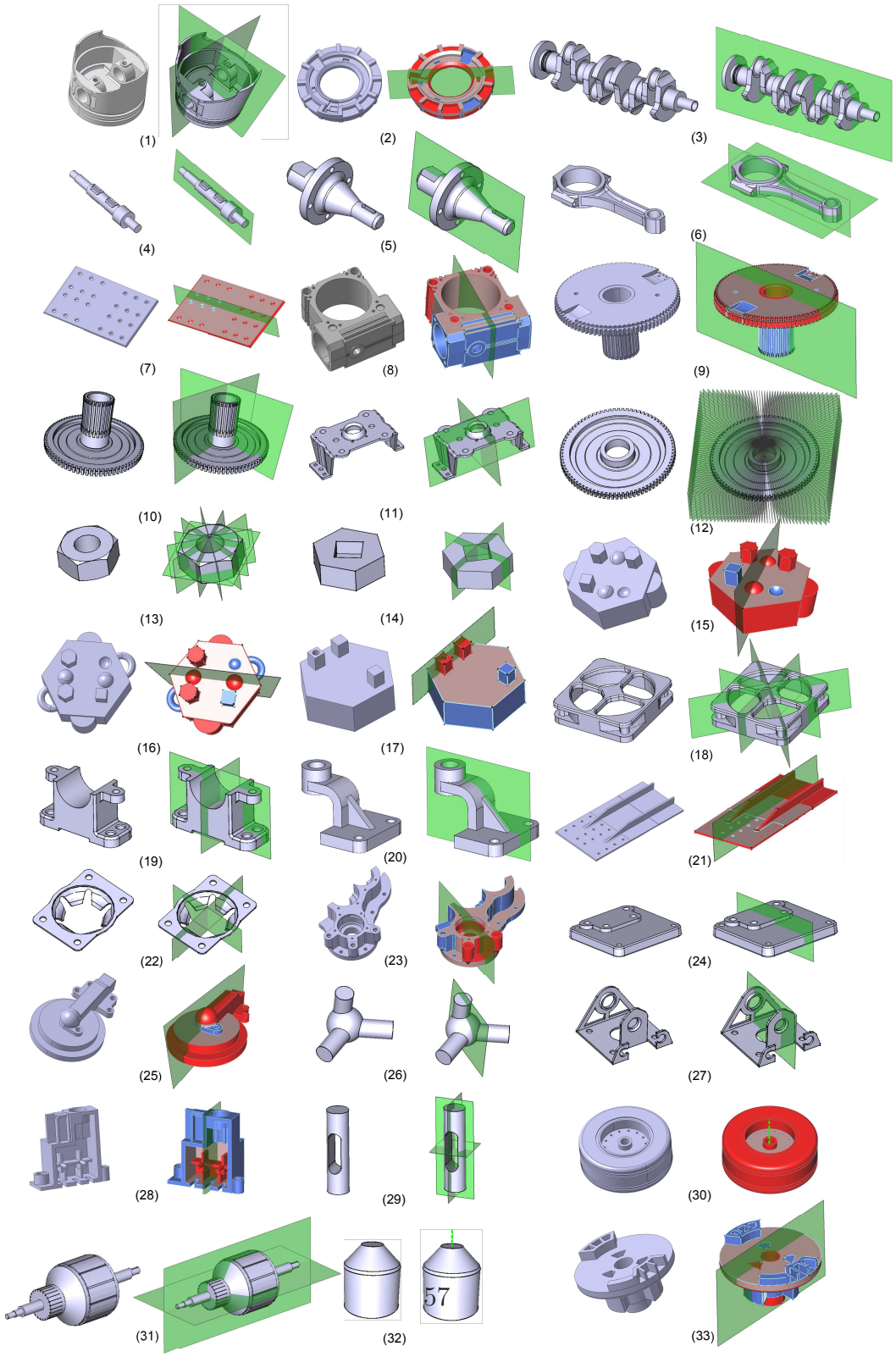


Figure 21: Results of global symmetry analysis on a repository of components. GPSs are highlighted for each of them if one at least exists. If none exists, one meaningful CSP has been selected with its extent highlighted.

An example of symmetric area in given in Figure 20a, b on the component of Figure 19. These areas are colored with the code described and they represent the areas obtained after the propagation process of first level. If a CSA exists, the same coloring principle holds for it and its icon is a dot-dash green line.

7.2. Analyzing the results on a set of components

On a larger scale, test cases have been processed on a repository of 45 mechanical components. Figure 21 highlights a subset of 33 components, some of them are similar to models in [10] and in [22]. For each model, the initial component is displayed aside to the result of the symmetry analysis. If one GSP at least exists, the object is colored in gray and all the GSPs found are highlighted. Otherwise, one CSP has been selected to display its corresponding extent of symmetric area. Each component is numbered so that it can be identified in Table 7.2 summarizing their geometric features (numbers of vertices, edges and faces of ∂M) and the effect of the generation of M_{MAX} (numbers of vertices, edges and faces of ∂M_{MAX}).

Timings of the generation of ∂M_{MAX}, i.e. vertex split, are given in Figure 22a since the other operators are of linear complexity. The results obtained show an approximated polynomial behavior of ${n_F}^{1.5}$. Timings of the division phase to assign the CSPs to ∂M_{MAX} are given in Figure 22b. The results obtained show an approximated linear behavior. In both cases, the complexity observed is clearly lower then the quadratic one obtained from the formal analysis.

Then, timings for the propagation phase are divided in accordance with the two phases involved with level one and level two propagations. Figure 23

Table 1: Parameters of the models analyzed in Figure 21.

Model nb	1	2	3	4	5	6	7	8	9	10	11
Faces	146	165	183	84	37	95	136	315	1426	953	190
Edges	423	466	447	218	85	265	356	754	4168	2760	462
Vertices	285	310	298	140	53	174	222	475	2750	1814	298
F_M	128	153	145	79	24	81	58	216	1199	810	143
E_M	341	419	284	186	37	212	96	472	3505	2348	286
V_M	212	310	148	112	12	132	16	256	2304	1540	140
Model nb	12	13	14	15	16	17	18	19	20	21	22
Faces	519	22	13	36	44	26	123	77	29	82	77
Edges	1534	48	30	86	106	60	320	197	81	264	222
Vertices	1022	28	20	58	70	40	200	122	54	200	148
F_M	425	11	13	34	36	25	123	66	25	50	70
E_M	1244	23	30	81	85	56	320	158	62	128	192
V_M	820	12	20	52	52	36	200	88	36	64	120
Model nb	23	24	25	26	27	28	29	30	31	32	33
Faces	236	86	110	11	72	143	8	82	197	19	184
Edges	611	218	306	25	196	406	24	192	582	36	489
Vertices	396	140	201	17	128	265	16	126	402	20	319
F_M	160	66	77	7	61	135	7	48	181	9	175
E_M	365	150	173	6	154	376	14	62	466	9	446
V_M	200	84	96	0	88	240	8	0	300	0	283

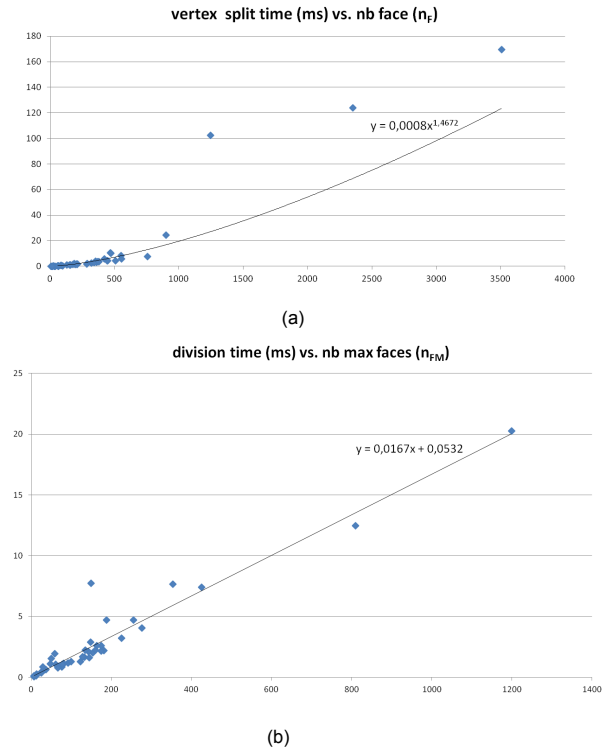


Figure 22: Timings for the vertex split operator used in the generation of ∂M_{MAX} (a) and for the division phase for the test components of the repository (b).

summarizes the results for the model repository. n_{F_M} is the reference parameter used to describe the shape complexity of a model. It is also the parameter used to describe the algorithmic complexity of the propagation phases. Figure 23a depicts the time evolution of the propagation process of level one. An acceptable fitting of the resulting point set is of the form $O({n_{F_M}}^{1.46})$. Similarly, the propagation of second level is characterized by Figure 23b using the same parameter n_{F_M} and the complexity obtained is of the form $O({n_{F_M}}^{1.77})$. As a result, each phase of the algorithm, sequentially from the generation of ∂M_{MAX} to the propagation of second level, has a polynomial complexity strictly lower than $O({n_{F_M}}^2)$. Overall, the complexity of the algorithm is polynomial of low order.

Figures 22 and 23 exhibit some outliers that correspond to models 9, 10, 12 in the repository (see Figure 21). These models contain significantly different features compared to others, i.e. number of internal loops per face and number of edges per face, which explains the relative difference of timings compared to others. This observation motivated the study of upper bounds of the algorithm's complexity using series of models with specific features and increasing in complexity.

7.3. Performance analysis under shape complexity variation

In order to scale up the model complexity and evaluate more precisely the upper bounds of the algorithm's complexity, some specific configurations and corresponding examples are proposed. First of all, faces bounded by many loops require more processing with LS-CSPs. This is addressed with a series of models containing one face bounded by an increasing number of loops (see Figure 24) obtained via protrusions modeled on one face. A first

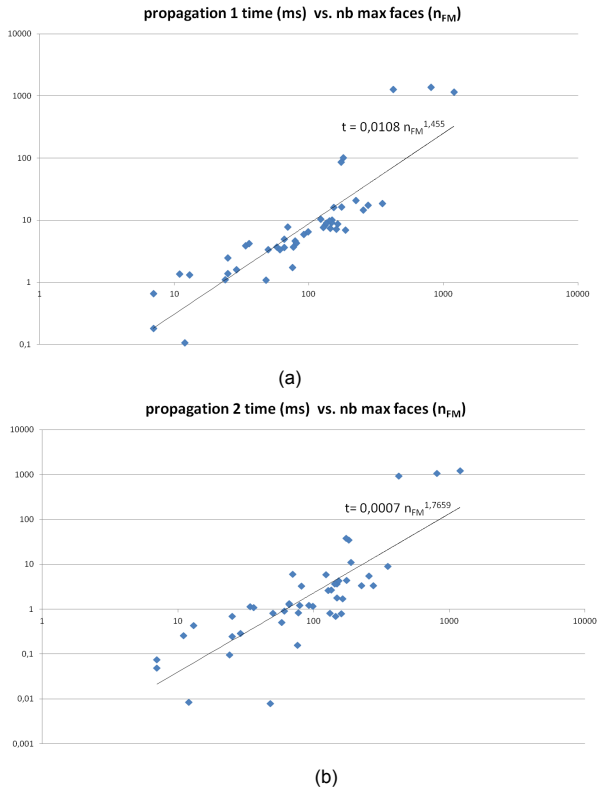


Figure 23: Timings for the propagation phases of first level (a) and second level (b) for the test components of the repository.

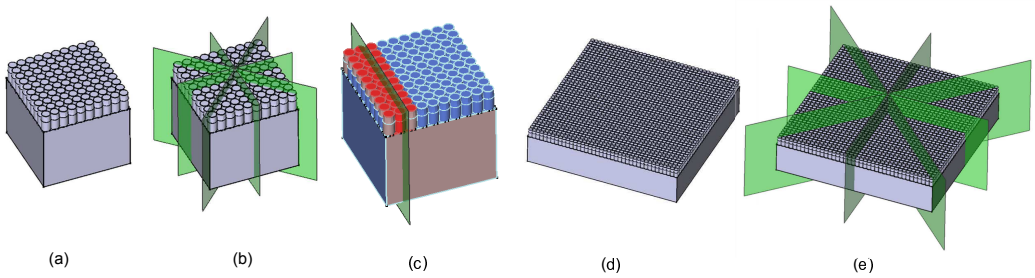


Figure 24: Samples of the series of models of increasing complexity with protrusions. a) simplest model with 100 protrusions, b) GSPs on model (a), c) an example of local symmetry on model (a), d) most complex model with 2500 protrusions, e) GSPs on model (c).

series contains 19 models having a number of loops on one face that ranges from 197, for the simplest model, up to 2501, for the most complex one, i.e. the number of loops equals the number of protrusions plus one. The number of faces of the input models ranges from 986 to 11526. The timing of the propagation phases for this series of models is given in Figure 25a, b. In this particular case, the complexity can be better estimated, as depicted by the fitting of the polynomial curve, and the exponent for the propagation of first level increases from 1.46 to 2.11 (see Figure 25a)). Indeed, it is the most constraining task compared to the propagation of second level (see Figure 25b).

The second series of models aims at evaluating more precisely the complexity of the propagation of second level. To this end, the models of the first series are modified so that the protrusions are scaled-up and overlapping each other, i.e. their diameter is larger than the axis distance between each protrusion. This series contains 20 models. As a result, the face with a large number of loops disappears and is replaced by a large number of faces bounded by a single loop. The simplest model has 1026 faces and the largest

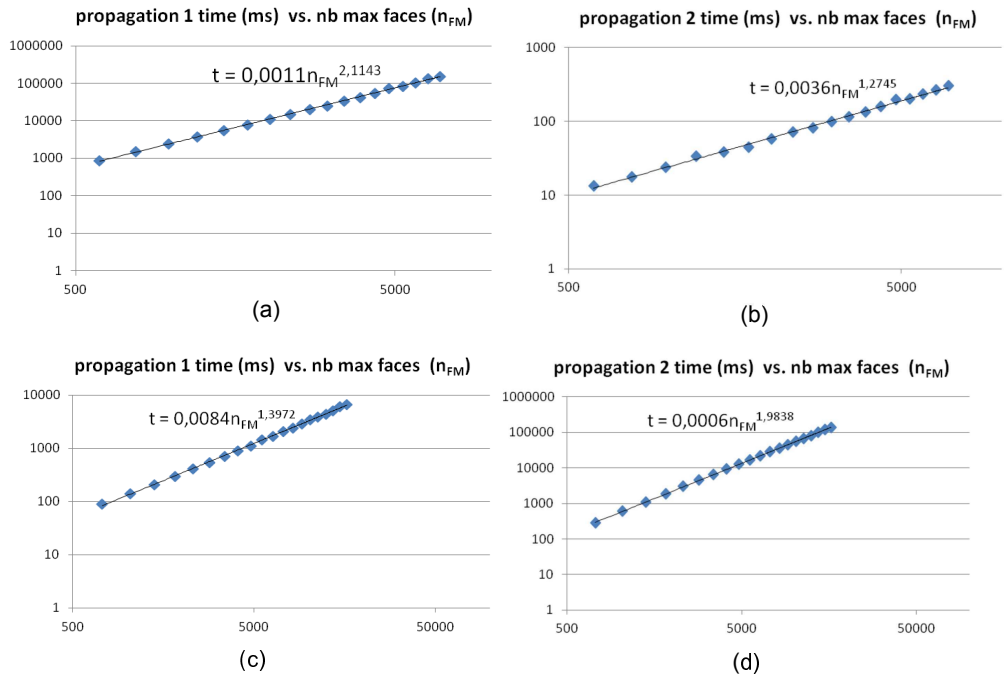


Figure 25: Timings for the propagation phases of the series of models with variable complexity. a) and b) propagation of first and second levels for the models with independent protrusions, c) and d) propagation of first and second levels for the models with interacting protrusions.

one, 23142. With these models, it is now possible to evaluate the complexity of the propagation of second level. As depicted in Figure 25d, the exponent for the propagation of second level increases from 1.77 to 1.99. This second task is effectively the most constraining one, as compared to the first one (see Figure 25c)). Overall, the upper bound of the propagation phase is slightly over quadratic. These series of models feature a large number of local CSPs and CSAs due to the 50x50 linear pattern features. These local symmetries are still available after generating the GSPs (see Figure 24c) and all the timings performed incorporate the generation of these local symmetries.

Also, the series of models show the evolution of the time to process highly

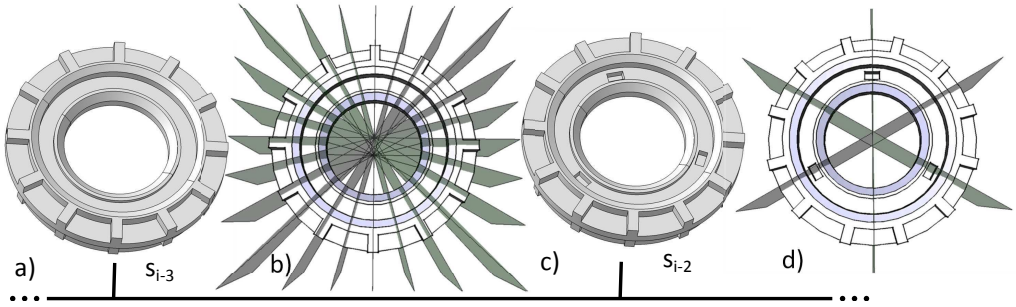


Figure 26: Complementary subset of construction tree displayed in addition to the subset displayed on Figure 4 to illustrate the impact of global symmetry properties over local ones.

complex models since the most complex models require 2mn 40s and 2mn 24s, for the first and second series, respectively. With regard to their complexity, in terms of number of faces, these times are acceptable.

7.4. Analyzing symmetry during a solid modeling process

Based on the proposed algorithm and on solid modeling processes using construction trees, Figure 26 shows a complement to Figure 4 with complementary steps of the same construction process where further global symmetry properties appear when modeling primitives are removed over time. These complementary steps enforce the interest of global symmetry analysis over local symmetry analyses that produce many more information that would be difficult to process because the extent of a local symmetry analysis seems difficult to adjust automatically.

8. Conclusion and future work

Taking a B-Rep CAD model as input, the proposed method can analyze the symmetry properties of a component at a global as well as a local level.

The proposed approach is independent of the parameterization of the faces and edges of the component boundary. The analysis is effectively intrinsic to the component shape and it is performed without requiring the geometry of edges. This is achieved thanks to the boundary decomposition generated using maximal faces and edges obtained using the merging operators as well as the split vertex one. Consequently, the reference surfaces describing the component boundary enable an accurate computation of its symmetry properties in a geometric modeling kernel. As a result, the symmetry properties extracted from the component can contribute to the generation of construction trees defining several ways this component can be obtained. As a main benefit of the proposed divide and conquer approach, these properties can be obtained in interactive time to help integrating them in a larger range of applications. The statistics highlight a complexity showing that the algorithm scales well for large and very large models.

Because of the thorough exploitation of the interactions between ∂M_{MAX} and the symmetry planes, all the GSPs and GSA are obtained. Local symmetries are derived in the same way and obtained simultaneously with the GSPs and GSA but some local symmetry properties have not been addressed. Indeed, this raises the question of the use of the symmetry properties to avoid computing quantities that are not useful for given applications. It is particularly the case when generating construction trees of components. Here, the interest symmetry properties connected to construction trees has been briefly highlighted. It is the purpose of future work to analyze interactions between construction trees and symmetry properties. The existence of repetitions of primitives is another category of interaction with symmetry properties and

construction trees that will be addressed.

Future work will also extend the approach to a larger range of reference surfaces to widen the object categories addressed. Rotational symmetry analysis is also part of future developments. Other developments will incorporate this approach in shape analysis, similarity and modeling processes of components.

Acknowledgments

The authors kindly thank the reviewers for their useful comments that significantly helped in describing the proposed approach.

References

[1] S. Parry-Barwick, A. Bowyer, Symmetry analysis and geometric modelling, in: Proceedings of DCTA93, Digital Image Computing-Techniques and Applications, pp. 39–46.

[2] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Symmetry descriptors and 3d shape matching, in: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 115–123.

[3] M. Leyton, A Generative Theory of Shape, Lecture Notes in Computer Science LNCS 2145, Springer-Verlag, 2001.

[4] D. Caillière, F. Denis, D. Pelé, A. Baskurt., 3d mirror symmetry detection using hough transform, in: Proceedings of ICIP08, pp. 1772–75.

[5] M. Bokeloh, M. Wand, H.-P. Seidel, A connection between partial symmetry and inverse procedural modeling, ACM Transaction on Graphics 29 (2010) 104–10.

[6] N. J. Mitra, M. Pauly, M. Wand, D. Ceylan, Symmetry in 3d geometry: Extraction and applications, in: STAR of Eurographics conference, pp. 29–51.

[7] Y. Wang, K. Xu, J. Li, H. Zhang, A. Shamir, L. Lui, Z. Cheng, Y. Xiong, Symmetry hierarchy of man-made objetcts, Computer Graphics Forum 30 (2011). Proceedings Eurographics 2011.

[8] M. Li, F. Langbein, R. Martin, Detecting design intent in approximate cad models using symmetry, CAD 42 (2010) 183–201.

[9] N. Mitra, L. Guibas, M. Pauly, Partial and approximate symmetry detection for 3d geometry, ACM Transaction on Graphics 25 (2006) 560–8.

[10] C. Gao, F. Langbein, A. Marshall, R. Martin, Approximate congruence detection of model features for reverse engineering, in: Proc. IEEE int. conf. shape modelling and applications, pp. 69–77.

[11] J.-C. Cuillière, V. Francois, J.-M. Drouet, Towards the integration of topology optimization into the cad process, Computer-Aided Design and Applications 11 (2014) 120–140.

[12] K. Sugihara, An $n \log n$ algorithm for determining the congruity of polyhedra, Journal of Computer and System Sciences 29 (1984) 36–47.

[13] J. Wolter, T. Woo, R. Volz, Optimal algorithm for symmetry detection in two and three dimensions, The Visual Computer 1 (1985) 37–48.

[14] P. Brass, C. Knauer, Testing congruence and symmetry for general 3-dimensional objects, Computational Geometry 27 (2004) 3–11.

[15] A. Martinet, C. Soler, N. Holzschuch, F. Sillion, Accurate detection of symmetries in 3d shapes, ACM Transactions on Graphics 25 (2006) 439–64.

[16] N. Mitra, L. Guibas, M. Pauly, Symmetrization, ACM Transactions on Graphics 63 (2007) 1–8.

[17] M. Bokeloh, A. Berner, M. Wand, H.-P. Seidel, A. Schilling, Symmetry detection using line features, Comp. Graphics Forum 28 (2009) 697–706.

[18] B. Mills, F. Langbein, A. Marshall, R. Martin, Approximate symmetry detection for reverse engineering, in: Proc. 6th ACM Symp. Solid and Physical Modeling, pp. 241–8.

[19] S. Iwanowski, Testing approximate symmetry in the plane is np-hard, Theo. Comp. Science 80 (1991) 227–62.

[20] M. Li, F. Langbein, R. Martin, Detecting approximate symmetries of discrete point subsets, CAD 40 (2008) 76–93.

[21] M. Li, F. Langbein, R. Martin, Constructing regularity feature trees for solid models, in: Lecture Notes in Computer Science LNCS 4077, GMP 2006, Springer Verlag, 2006, pp. 267–286.

[22] J. Jiang, Z. Chen, K. He, A feature-based method of rapidly detecting global exact symmetries in cad models, CAD to appear (2013).

[23] V. Shapiro, D. Vossler, Separation for boundary to csg conversion, ACM Trans. Graphics 12 (1993) 35–55.

[24] Y. Woo, H. Sakurai, Recognition of maximal features by volume decomposition, CAD 34 (2002) 195–207.

[25] Y. Woo, Fast cell-based decomposition and applications to solid modeling, CAD 35 (2003) 969–977.

[26] M. Li, F. Langbein, R. Martin, Detecting approximate incomplete symmetries in discrete point sets, in: ACM Symp. Solid and physical modeling, pp. 335–40.

[27] S. Cohen, G. Elber, R. Bar-yehuda, Matching of freeform curves, CAD 29 (1997) 369–378.

[28] R. T. Farouki, Optimal parameterizations, CAGD 14 (1997) 153–168.

[29] Y. Ma, Y. Lee, Detection of loops and singularities of surface intersections, CAD 30 (1998) 1059–1067.

[30] H. Park, K. Kim, S.-C. Lee, An method for approximate nurbs curve compatibility based on multiple curve refitting, CAD 32 (2000) 237–252.

[31] Y.-J. Yang, C.-L. Yang, B. Deng, X.-X. Meng, W. Zeng, An algorithm to improve parameterizations of rational bzier surfaces using rational bilinear reparameterization, CAD 44 (2012).

[32] S. Tate, G. Jared, Recognising symmetry in solid models, CAD 35 (2003) 673–92.

[33] B. Mills, F. Langbein, A. Marshall, R. Martin, Estimate of frequencies of geometric regularities for use in reverse engineering of simple mechanical components, Tech. report GVG 2001-1, Dept. Computer Science, Cardiff University, 2001.

[34] M. K. Agoston, Computer Graphics and Geometric Modeling, Mathematics, Springer Verlag, 2005.

[35] A. Sheffer, T. Blacker, J. Clements, M. Bercovier, Virtual topology operators for meshing, in: 6th International Meshing Roundtable.

[36] A. Sheffer, Model simplification for meshing using face clustering, CAD 33 (2001) 925–934.

[37] G. Foucault, J.-C. Cuillière, V. Francois, J.-C. Léon, R. Maranzana, Adaptation of cad model topology for finite element analysis, CAD 40 (2008) 176–196.

[38] C. E. Silva, Alternative definitions of faces in Boundary representations of solid objects, Technical Report, Carnegie Mellon University, 1981.

[39] A. A. G. Requicha, Representations for rigid solids: theory, methods, and systems, ACM Computing Surveys 12 (1980) 437–464.

[40] F. Boussuge, J.-C. Léon, S. Hahmann, L. Fine, Extraction of generative processes from b-rep shapes and application to idealization transformations, CAD 46 (2014) 79 – 89.

[41] M. Frigo, S. G. Johnson, Manual for the library 'fastest fourier transform in the west' (fftw version 3.3.3) (http://www.fftw.org/fftw3.pdf), 2013.