Consiglio Nazionale delle Ricerche
Istituto di Matematica Applicata
e Tecnologie Informatiche
"Enrico Magenes"

REPORT SERIES

L. Chiang, F. Giannini, M. Monti

Identification of Patterns of Repeated Parts in Solid Objects
Part II

16-15

IMATI REPORT Series

Nr. 16-15 – November 2016

IMATI Report Series Nr. 16-15
2nd November 2016

Identification of Patterns of Repeated Parts in Solid Objects
Part II

L. Chiang, F. Giannini, M. Monti

Corresponding author: franca.giannini@ge.imati.cnr.it

Abstract.

The work described in this report extends the methods described in the report "Identification of Patterns of Repeated Parts in Solid Objects" ([2]) to deal with CAD assemblies and to the detection of additional types of patterns. Specifically rotational and compound patterns of repeated elements are treated. Therefore, references of all notations, definitions and terms of the objects cited in this report can be found in [2]. The Chapter 1 describes the proposed approach to detect rotational patterns of repeated entities in part model, extending the set of types of patterns considered in [1], [2]. Chapter 2 contains the method proposed for assembly models for the detection of symmetric arrangements on components, considering as repeated elements of the patterns the instances of a part or assembly file. Chapter 3 is dedicated to the introduction of the concept of compound pattern, intended as an improvement of the quality of the symmetry information of a model, obtained by combining the information referred to the patterns of repeated elements. The approach has been described and provided either for part and assembly models.

Keywords: *Regular pattern detection, BRep models, shape analysis*

Contents

Chapter 1

Patterns for part models

1.1 Rotation

In this section we introduce and define the concept of rotational pattern of repeated sub-parts in part model. This pattern can be added to the set of patterns that can be detected by the *pattern detection algorithm* [1], [2], inserting a procedure adressed to the detection of this new type of pattern in case of circular path of centroids.

We first introduce the theoretical notions related to rotation and then describe the requirements that the REs should fulfill to represent a rotational pattern.

1.1.1 Theory of rotation

Every rotation of a solid object can be expressed by a *rotation matrix* [3]. The rotation is realized through the characterization of the solid object with a local coordinate system associated to it. Suppose the local coordinate system to be $(Oxyz)$, the *rotation axis* to have coordinates (r_x, r_y, r_z), with $\sqrt{r_x^2 + r_y^2 + r_z^2} = 1$ (i.e. the vector $\overrightarrow{r}$ is a unit vector). The rotation of the object with respect to an angle θ ($\theta > 0$) and in counterclockwise with respect to an axis passing through O and with corresponding direction $\overrightarrow{r}$ is the result of the composition of elementary rotations (see [3]) and the corresponding matrix is:

$$R_{r,\theta} = \begin{bmatrix} r_x^2(1-C_\theta)+C_\theta & r_xr_y(1-C_\theta)-r_zS_\theta & r_xr_z(1-C_\theta)+r_yS_\theta \\ r_xr_y(1-C_\theta)+r_zS_\theta & r_y^2(1-C_\theta)+C_\theta & r_yr_z(1-C_\theta)-r_xS_\theta \\ r_xr_z(1-C_\theta)-r_yS_\theta & r_yr_z(1-C_\theta)+r_xS_\theta & r_z^2(1-C_\theta)+C_\theta \end{bmatrix}, \tag{1.1}$$

where C_θ denotes $\cos\theta$ and S_θ denotes $\sin\theta$.

In practice, for every point P of the considered solid object with coordinates (x_P, y_P, z_P) the rotated of P with respect to an angle θ and an axis direction $\overrightarrow{r}$ has coordinates defined by:

$$R_{r,\theta}(x_P, y_P, z_P)^T \tag{1.2}$$

1.1.2 Rotational pattern detection

Let $\{\mathbf{A}_0, \ldots, \mathbf{A}_{q-1}\}$ be a set of REs and $\{C_0, \ldots, C_{q-1}\}$ be the set of the respective centroids, $q \geq 2$. Suppose $(C_0, \ldots, C_{q-1})$ to be a path.

We will suppose $\mathbf{F}_j^{\mathbf{p}}$ and $\mathbf{F}_j^{\mathbf{c}}$ to be respectively the set of planar faces and the set of cylindrical faces of $\mathbf{A}_j$, for $j = 0, \ldots, q-1$. The congruency of the REs implies that $|\mathbf{F}_j^{\mathbf{p}}| = |\mathbf{F}_k^{\mathbf{p}}|$ and $|\mathbf{F}_j^{\mathbf{c}}| = |\mathbf{F}_k^{\mathbf{c}}|$ for $j, k = 0, \ldots, q-1$.

Suppose $(C_0, \ldots, C_{q-1})$ be a path of q centroids of type circumference (see [1], [2]). To conclude that the set of q REs $\{\mathbf{A}_0, \ldots, \mathbf{A}_{q-1}\}$ is a *rotational pattern* it is necessary to examine the REs in pairs.

The computation of the *candidates* rotational angle and rotational axis direction for the current circular path must be performed before the start of the check steps and it is made in the following way. Let $C_0 = (x_{C_0}, y_{C_0}, z_{C_0})$ and $C_1 = (x_{C_1}, y_{C_1}, z_{C_1})$. Recall the dot product formula:

$$a \cdot b = \|a\|\|b\| \cos\theta.$$

The candidate rotational angle is computed as the angle covered moving a centroid in the path to its consecutive, for example from the centroid C_0 to C_1, in the plane of the circumference lies in and considering as rotation axis the one along the plane normal passing through the centre of the circumference. Let denote with $C = (x_C, y_C, z_C)$ the centre of the path circumference, then we define $v_{C_0,C} = (x_{C_0} - x_C, y_{C_0} - y_C, z_{C_0} - z_C)$ and $v_{C_1,C} = (x_{C_1} - x_C, y_{C_1} - y_C, z_{C_1} - z_C)$. The candidate rotational angle θ is given by:

$$\theta = \arccos\left(\frac{v_{C_0,C} \cdot v_{C_1,C}}{\|v_{C_0,C}\|\|v_{C_1,C}\|}\right).$$

Observe that the centroids are translated by the vector $(-x_C, -y_C, -z_C)$ as the rotation matrix considers the rotation axis passing through the origin of $\mathbb{R}^3$ (denoted by O), so the entire coordinate system must be translated such that the axis direction passes through C.

The axis direction must be defined such that the resulting rotation is made in counterclockwise sense respect to the outer axis direction. This vector is established by using the vector product: $v_{C_0,C} \times v_{C_1,C}$ is an outer vector respect to the plane generated by $v_{C_0,C}, v_{C_1,C}$ according to the right hand rule. The candidate rotational axis direction also corresponds to the direction of the normal vector to the plane containing the path circumference, up to a multiplying constant -1. Let be $\overrightarrow{r}$ the candidate axis direction.

The candidate rotation matrix is $R_{r,\theta}$ (see Equation 1.1).

Let consider two REs $\mathbf{A}_i, \mathbf{A}_{i+1}$ with consecutive centroids C_i, C_{i+1} in the given path , $i \in \{0, \ldots, q-2\}$.

To conclude that $\mathbf{A}_i, \mathbf{A}_{i+1}$ are related by rotation according to the candidate rotation matrix $R_{r,\theta}$ we provide a two-level check, the first considers the vertices, the second exploits the surface information of the faces.

1. *Vertex check.* For each vertex $V_{i,k}$, $k \in \{0, \ldots, p-1\}$, we must check if there

exist $h \in \{0, \dots, p-1\}$ such that

$$v_{V_{i+1,k},C} = R_{r,\theta}\, v_{V_{i,k},C}^T.$$

If the condition is satisfied, we apply the next step.

2. *Face check*. Different checks are provided for planar faces and for cylindrical faces.

   - *Planar faces*. For each planar face we compare the normals. We simply verify if for each planar face $F_k \in \mathbf{F}_i^{\mathbf{p}}$ there exist a planar face $F_h \in \mathbf{F}_1^{\mathbf{p}}$ such that

     $$n_h = R_{r,\theta}\, n_k^T,$$

     where n_k, n_h are the normal vectors of the planes corresponding respectively to F_k, F_h.

   - *Cylindrical faces*. For each cylindrical face in $\mathbf{A}_i$ such that it has at least a closed planar base edge we check if there exist a corresponding rotated cylindrical face in $\mathbf{A}_{i+1}$ by exploiting axis and edges information.

     Let $F_k \in \mathbf{F}_i^{\mathbf{c}}$, $\mathcal{C}_k$ the underlying cylinder, a_k the direction vector of the axis of $\mathcal{C}_k$, r_k its radius, O_k its origin, $\mathbf{W}_k$ the set of the circumferences corresponding to its closed planar base edges, $\mathbf{Z}_k$ the set of the ellipses corresponding to its closed planar base edges (information provided by the adopted CAD system), such that $|\mathbf{W}_k| + |\mathbf{Z}_k| > 0$.

     We select the set of cylindrical faces

     $$\mathbf{C}_{i+1} = \{F_h \in \mathbf{F}_{i+1}^{\mathbf{c}} : r_h = r_k, |\mathbf{W}_h| = |\mathbf{W}_k|, |\mathbf{Z}_h| = |\mathbf{Z}_k|\},$$

     where r_h, $\mathbf{W}_h$, $\mathbf{Z}_h$ respectively denote the radius, the set of circumferences corresponding to the closed planar base edges, the set of ellipses corresponding to the closed planar base edges of the cylindrical face F_h.

We select from $\mathbf{C}_{i+1}$ the set of cylindrical faces with axis corresponding to the line computed in the following way. Let G be the point whose coordinates are defined by $v_{G,C} = R_{r,\theta}\ v^T_{O_k,C}$, it should be on the axis line of the candidate cylinder. The cylindrical face we are looking for has axis corresponding to the line computed by considering the line passing through G with direction $R_{r,\theta}\ a_k$, so we select from in $\mathbf{C}_{i+1}$ the cylindrical faces satisfying this condition.

When a face F_h satisfying these conditions is found, we finally examine the base edges correspondence. For each circumference in $\mathbf{W}_k$ with centre C^c_k, we check if there exist in $\mathbf{W}_h$ a circumference with centre C^c_h such that $v_{C^c_h,C} = R_{r,\theta}\ v^T_{C^c_k,C}$; for each ellipse in $\mathbf{Z}_k$ with centre C^e_k, we check if there exist in $\mathbf{Z}_h$ an ellipse with centre such that $v_{C^c_h,C} = R_{r,\theta}\ v^T_{C^c_k,C}$.

Furthermore, for each ellipse in $\mathbf{Z}_k$ with v_{major} and v_{minor} as direction vector of respectively the major and the minor ellipse axis, the corresponding ellipse in F_h must have $R_{r,\theta}\ v_{major}$ and $R_{r,\theta}\ v_{minor}$ respectively as direction vector of respectively the major and the minor ellipse axis.

If a cylindrical face F_h satisfying all the cited conditions exists, it corresponds to the rotation of the cylindrical face F_k.

If both vertex and face checks give a positive answer, we can state that $\mathbf{A}_i$ and $\mathbf{A}_{i+1}$ are related by a rotation with rotational matrix $R_{r,\theta}$.

Finally, we can say that the set of REs $\{\mathbf{A}_0, \ldots, \mathbf{A}_{q-1}\}$ constitute a *rotational pattern* if $\mathbf{A}_i$ and $\mathbf{A}_{i+1}$ are related by rotation with the rotation matrix $R_{r,\theta}$, for $i = 0, \ldots, q-2$.

1.2 Tests for rotational pattern in part model

In this section we present the results obtained applying the developed functions for the rotational pattern detection in C♯ on two test part models. The models are embedded in the CAD system SolidWorks© and the generated methods are recalled as a plug-in in it (for more details see [1], [2]).

1.2.1 Gear

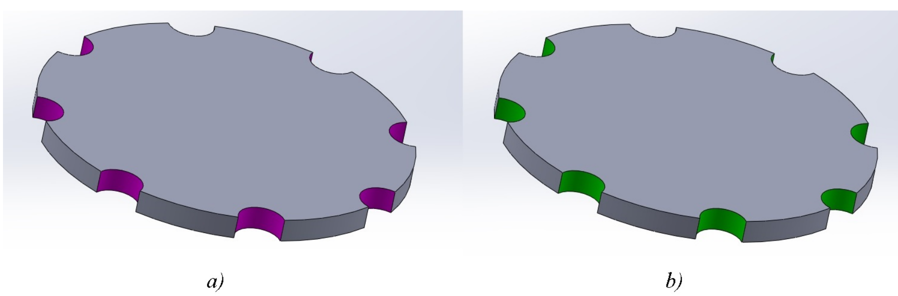


Figure 1.1: Gear. a) Selected REs; b) detected pattern.

We consider the gear part model in Figure 1.1. Figure 1.1 a) highlights the selected cylindrical faces as input (in violet), constituting the REs whose any regular arrangement is searched.

The algorithm detects a single rotational pattern (Figure 1.1 b), constituted by all the 8 initially selected REs.

1.2.2 Double gear

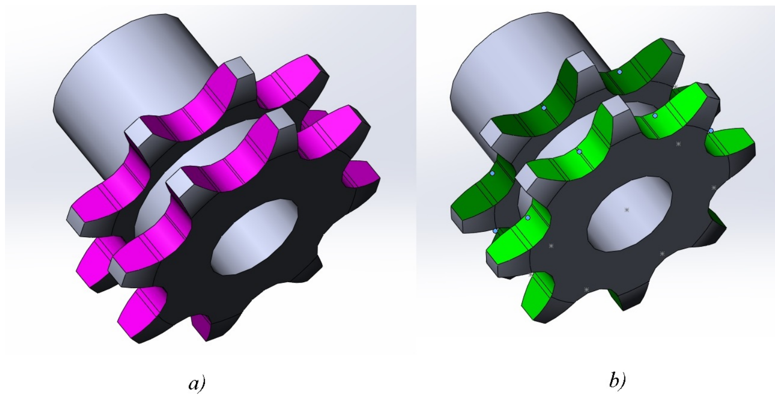


Figure 1.2: Gear. a) Selected REs; b) detected patterns.

The input set in this example is constituted by a set of 16 REs, each of them composed by 4 cylindrical faces and 3 planar faces (the selection is represented in Figure 1.2 a).

The algorithm detects 2 rotational patterns (Figure 1.2 b), each of them constituted by the 8 of the initially selected REs. The circumferences corresponding to the patterns lies on parallel plane and they have centers lying on the a line with direction orthogonal to the planes.

Chapter 2

Pattern detection in assembly models

The proposed method has also been extended to the case of *assembly models*. This class of CAD models aims at handling multiple files representing components within an object. These files may be part models or even other assembly models (sub-assembly) compound in turn by other assemblies or parts. In the following, the instances of the cited files recalled in the model will be called simply *components* of the assembly model. Every instance (part or assembly) component of another one is said to be its *son*, while the grouping component is the *father*.

The API of SolidWorks© provides specific functions created to accurately describe the position of a specific component respect to its father. This description consists on a 4×4 *transformation matrix* containing information referred to the position of the local coordinate system associated to the instance. In particular, a 3×3 sub-matrix represents the rotation or reflection and an array of length 3 describes the translation applied to position the instance within the assembly. Furthermore, the provided matrix reports a scalar value representing the scaling factor of the instance respect to the original file.

The total (absolute) transformation matrix (referred to the position of the component respect to the origin file) can be easily obtained by composing the transformation matrix provided by API with the transformation matrix of the father.

The pattern search procedure for assembly models is very similar to the one proposed for repeated sub-parts in part models. In the current development status the repetitions are represented by the sets of instances of a specified file. For this reason the starting step is no more constituted by the user intervention (selection of the faces in the model) but the process starts the detection automatically, simply scanning and analysing the sets of instances of each component.

Another difference with respect to the part approach consists on skipping the grouping surface step, since information on the faces in contact between components might be not available. To make the process faster consider a "reference coordinate system based" process, which means that we do not want to examine the details of the component structures, but we choose to consider only the orientation of the objects to make conclusions about their arrangement.

2.1 Representative point identification

A preliminary phase based on point position analysis is performed, analogously to the part approach. This is done by choosing the *origin* of the local coordinate system associated to each component as representative point for it. Such a point can be easily obtained by considering as its coordinates the vector (x_v, y_v, z_v), which defines the translational vector v reported by the total transformation matrix of the component instance.

Consider the set $\{I_0, \ldots, I_{n-1}\}$ of the instances referred to a fixed component (part or assembly) and the set of corresponding origins $\{O_0, \ldots, O_{n-1}\}$ obtained in the just described way. Then, the *adjacency matrices at constant distance* are computed on the origin set and the *path detection algorithm* is applied on them, analogously to the procedure described in [1], [2]. The adjacency matrices are reordered by creasing ordering respect to the repeated distance of d. For a fixed adjacency matrix a set of *paths of points* (of origins) is detected.

2.2 Patterns for assembly models

As already stated in [1] and [2], a path of representative points for the considered repeated components (we will simply call them RCs) gives an indication of the placement of the instances suggesting candidate regular arrangements. Therefore, verifying the correct orientation only for the RCs whose origins are regularly organized, optimises the number of checks. In this section the orientation analysis of these components is described according to the type of patterns considered in our work.

Let $\{\mathbf{I}_0, \ldots, \mathbf{I}_{q-1}\}$ be a set of RCs and $\{O_0, \ldots, O_{q-1}\}$ be the set of the respective origins, $q > 2$. Suppose $(O_0, \ldots, O_{q-1})$ to be a path.

If $(O_0, \ldots, O_{q-1})$ is a path of type line, the RCs in $\{\mathbf{I}_0, \ldots, \mathbf{I}_{q-1}\}$ could constitute a pattern of type:

- linear translational;

If $(O_0, \ldots, O_{q-1})$ is a path of type circumference, the RCs in $\{\mathbf{I}_0, \ldots, \mathbf{I}_{q-1}\}$ could constitute a pattern of type:

- circular translational;
- rotational.

The existence of a pattern of components is established by exclusively analysing the local coordinate system associated to every instance, which completely defines the component position in the space.

The three unit vectors defining such a coordinate system associated to a component are the column vectors of the 3×3 sub-matrix of the transformation matrix associated to it. So, denoting with T_{I_i} the transformation matrix of the component I_i, $i \in \{0, \ldots, q-1\}$, the local coordinate system of I_i is defined by the three vectors:

$$\overrightarrow{v_{x,I_i}} = (T_{I_i}(0,0), T_{I_i}(1,0), T_{I_i}(2,0))$$

$$\overrightarrow{v_{y,I_i}} = (T_{I_i}(0,1), T_{I_i}(1,1), T_{I_i}(2,1))$$

$$\overrightarrow{v_{z,I_i}} = (T_{I_i}(0,2), T_{I_i}(1,2), T_{I_i}(2,2))$$

2.2.1 Linear translational pattern

Suppose $(O_0, \ldots, O_{q-1})$ be a path of type line. To conclude that the set of RCs $\{\mathbf{I}_0, \ldots, \mathbf{I}_{q-1}\}$ is a *linear translational pattern* it is necessary to examine the RCs in pairs.

Let $O_i = (x_{O_i}, y_{O_i}, z_{O_i})$ and $O_{i+1} = (x_{O_{i+1}}, y_{O_{i+1}}, z_{O_{i+1}})$. Then, the candidate *translational vector* is $w = (x_w, y_w, z_w)$ computed as

$$(x_{O_{i+1}} - x_{O_i}, y_{O_{i+1}} - y_{O_i}, z_{O_{i+1}} - z_{O_i}). \tag{2.1}$$

Observe that the candidate translational vector is the same for every pair of consecutive RCs, since their origins lie on a detected path.

Consider two RCs $\mathbf{I}_i, \mathbf{I}_{i+1}$ with consecutive origins O_i, O_{i+1} in the given path , $i \in \{0, \ldots, q-2\}$.

To conclude that $\mathbf{I}_i, \mathbf{I}_{i+1}$ are related by translation by the candidate translational vector w we require:

$$\overrightarrow{v_{x,I_i}} = \overrightarrow{v_{x,I_{i+1}}}$$

$$\overrightarrow{v_{y,I_i}} = \overrightarrow{v_{y,I_{i+1}}}$$

$$\overrightarrow{v_{z,I_i}} = \overrightarrow{v_{z,I_{i+1}}},$$

and so $\mathbf{I}_{i+1} = T_w(\mathbf{A}_{i+1})$.

Finally, we can say that the set of RCs $\{\mathbf{I}_0, \ldots, \mathbf{I}_{q-1}\}$ constitutes a *linear translational pattern* if $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$ are related by a translational function with the translational vector w, for $i = 0, \ldots, q-2$.

2.2.2 Circular translational pattern

Suppose $(O_0, \ldots, O_{q-1})$ to be a path of centroids of type circumference. To conclude that the set of RCs $\{\mathbf{I}_0, \ldots, \mathbf{I}_{q-1}\}$ is a *circular translational pattern* it is necessary to examine the RCs in pairs, analogously to the translational pattern on line case.

As in the procedure described in [1], [2] for repeated sub-parts in part models, the candidate translational vector changes direction for each pair of consecutive RCs, while the euclidean norm of the vector is a constant value. Let $i, j \in \{0, \ldots, q-2\}$ and let $\mathbf{I}_i, \mathbf{I}_{i+1}$ and $\mathbf{I}_j, \mathbf{I}_{j+1}$ be two pairs of consecutive RCs. Then the candidate translational vector of the pair $\mathbf{I}_i, \mathbf{I}_{i+1}$, noted by w_i, is computed as $(x_{O_{i+1}} - x_{O_i}, y_{O_{i+1}} - y_{O_i}, z_{O_{i+1}} - z_{O_i})$, and analogously the candidate translational vector of the pair $\mathbf{I}_j, \mathbf{I}_{j+1}$, noted by w_j, is computed as $(x_{O_{j+1}} - x_{O_j}, y_{O_{j+1}} - y_{O_j}, z_{O_{j+1}} - z_{O_j})$ (see Formula 3.2). Denoting the euclidean norm of a vector $w \in \mathbb{R}^3$ as $\|w\|$, we have $\|w_i\| = \|w_j\| = d$, where d is the repeated distance between origins involved in the path of type circumference.

To conclude that the set of RCs $\{\mathbf{I}_0, \ldots, \mathbf{I}_{q-1}\}$ constitute a *circular translational pattern* we must verify that for $i = 0, \ldots, q-2$ the RCs $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$ are related by translational function with the translational vector w_i (as described in Subsec. 2.2.1).

2.2.3 Rotational pattern

Suppose $(O_0, \ldots, O_{q-1})$ to be a path of centroids of type circumference. To conclude that the set of RCs $\{\mathbf{I}_0, \ldots, \mathbf{I}_{q-1}\}$ is a *rotational pattern* it is necessary to examine the RCs in pairs, analogously to the translational pattern case.

The candidate rotation matrix is obtained in the way described in Subsection 1.1.2, considering origins instead of centroids in the procedure. So, we call $R_{r,\theta}$ the found candidate rotation matrix.

Consider two RCs $\mathbf{I}_i, \mathbf{I}_{i+1}$ with consecutive origins O_i, O_{i+1} in the given path , $i \in \{0, \ldots, q-2\}$.

To conclude that $\mathbf{I}_i, \mathbf{I}_{i+1}$ are related by rotation by the candidate rotation matrix $R_{r,\theta}$ we require:

$$v_{x,I_{i+1}} = R_{r,\theta}\, v_{x,I_i}^T$$

$$v_{y,I_{i+1}} = R_{r,\theta}\, v_{y,I_i}^T$$

$$v_{z,I_{i+1}} = R_{r,\theta}\, v_{z,I_i}^T$$

Finally, we can say that the set of RCs $\{\mathbf{I}_0, \ldots, \mathbf{I}_{q-1}\}$ constitute a *rotational pattern* if $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$ are related by rotation with the rotation matrix $R_{r,\theta}$, for $i = 0, \ldots, q-2$.

Observation 2.2.1. *Reflectional patterns ($q = 2$) are not considered in assembly case. In fact, the reflected of a component is no more the component itself, as it has been subjected to an orientation reversing isometry.*

2.3 Pattern detection algorithm for assembly models

In this section we illustrate the sequence of the steps of the *pattern detection algorithm in assembly models*. We provided an outline of the method in the previous sections. The "*FIND_ALL_PATHS*" procedure recalled in the main algorithm and constituting a fundamental part of the process corresponds to the *path detection algorithm* widely described in [1], [2], therefore for its details the reader should refer [2].

Here follows the pseudo-code that summarizes the entire algorithm.

TRAVERSE_ASSEMBLY_COMPONENT

Input: Component, Instance_list (list of lists of instances)

Output: Instance_list

\\ *Get the children components of the input component*

Children_list = *GET_CHILDREN(*Component*)*

for each *(*$\mathcal{C}$ *in* Children_list*)* ***do***

  \\ *Get the transformation matrix of this children component*

  GET_TRANSFORM_MATRIX ($\mathcal{C}$)

  \\ *Update the list of lists of instances*

  UPDATE_LIST_OF_INSTANCES ($\mathcal{C}$, Instance_list)

  \\ *If the children component is not a "leaf" (part), apply the current function to it*

  if IS_LEAF($\mathcal{C}$) ***then***

    Instance_list=*TRAVERSE_ASSEMBLY_COMPONENT (*$\mathcal{C}$*)*

  end if

end for

PATTERN_DETECTION_FOR_ASSEMBLY

Input: Assembly

Output: Pattern_list, list of patterns of RCs

Pattern_list = *empty*

Pattern_length2_list = *empty*

Instance_list=*TRAVERSE_ASSEMBLY_COMPONENT (*Assembly*)*

THIN_OUT_AND_ORDERING(Instance_list)

for each *(*instances *in* Instance_list*)* ***do***

  List_adjacency_matrices = *COMPUTE_ADJACENCY_MATRICES (*instances*)*

  THIN_OUT_AND_ORDERING(List_adjacency_matrices)

  for each *(*$\mathcal{M}$ *in* List_adjacency_matrices*)* ***do***

    List_of_paths = *FIND_ALL_PATHS* ($\mathcal{M}$)

    ORDERING(List_of_paths)

for each *(*$\mathcal{P}$ *in* List_of_paths*)* ***do***

\\ *Verification of the identified paths*

VERIFY_PATTERNS_AND_UPDATE_LIST_OF_PATTERNS(

$\mathcal{P}$, Pattern_list, Pattern_list_length2)

UPDATE_OTHER_DATA(

List_of_paths,List_adjacency_matrices, Pattern_list_length2)

Remove $\mathcal{P}$ *from* List_of_paths

end for

Remove $\mathcal{M}$ *from* List_adjacency_matrices

end for

Remove instances *from* Instance_list

end for

\\ *Final attempt to find patterns of length 2*

ARRANGEMENTS_IN_PATTERNS_OF_LENGTH2 (

Pattern_list_length2, Pattern_list)

return Pattern_list

In details, the operations implemented to reach the detection of symmetric arrangements of RCs are the followings.

1. *Acquisition of model.* The entire assembly model is acquired as input.

2. *Decomposition of the model.* The input model is decomposed in all its children components (part or assembly components) and they are grouped in coherent sets, such that each set contains all the instances of the file.

3. *Thin out and ordering of instance list.* All the sets containing only one element are deleted from the list of instances. Then, the remaining sets are ordered by descending criteria respect to the number of elements in the associated RCs list.

4. *Examination of the instance list and patterns verification*. Then, taking the first available set of instances from the list, the adjacency matrices are computed, the path detection algorithm applied on every adjacency matrix and the found paths geometrically verified by the examination of the orientation of the local coordinate systems (described in Section 2.2). This step is analogous to the corresponding in the approach proposed for repeated entities in part models [1], [2].

5. *Final attempt to find patterns of length* 2. At the end of the previous step there could remain a set of instances not yet in a detected symmetric arrangements of length at least 3. For these RCs we decided to associate them, if it exists, to an already found symmetric arrangement of length 2 containing them (these patterns are found "by chance" while we were looking for longer patterns, as explained in details in [1], [2]). If such a pattern of length 2 does not exist, a possibility (not implemented yet) is to try to couple the remaining RCs corresponding to the same origin file two by two, verifying if they are related by a regular arrangements (translation or rotation), starting the coupling attempting from the nearest couples of RCs. The examined instance list is then deleted.

2.4 Results for pattern detection in assembly models

In this section we present some of the results obtained applying the developed functions for the pattern detection on assembly models. The functions are developed in C$\sharp$ as a plug-in of the commercial CAD system SolidWorks$^{©}$ and exploit its API to access the assembly data.

2.4.1 Catapult

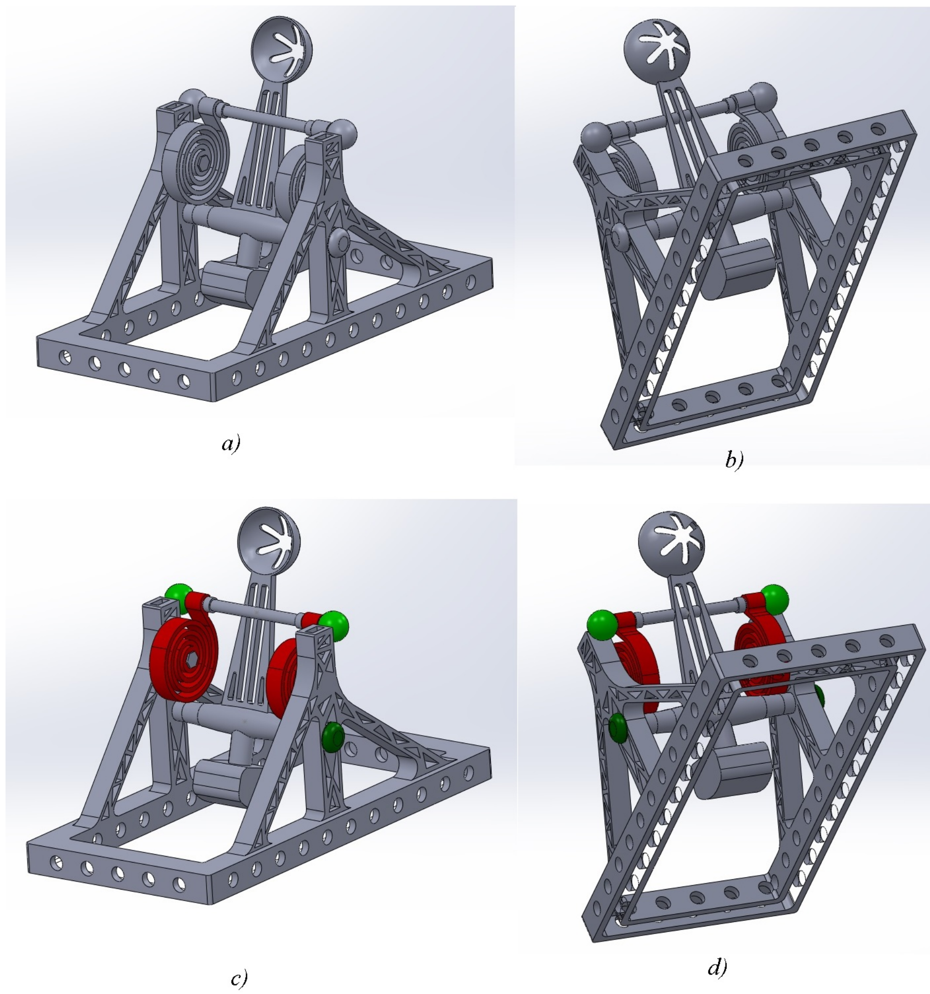


Figure 2.1: Catapult. a) Original model view 1; b) Original model view 2; c) detected patterns view 1; d) detected patterns view 2.

We consider the assembly model in Figure 2.1 a) and b).

The algorithm considers only the components whose instances are more than 1. In this situation it detects a translational pattern constituted by the two red components and

2 rotational patterns of 2 elements, each of them coloured in a green nuance (Figure 2.1 c) and d)).

2.4.2 Steel machine

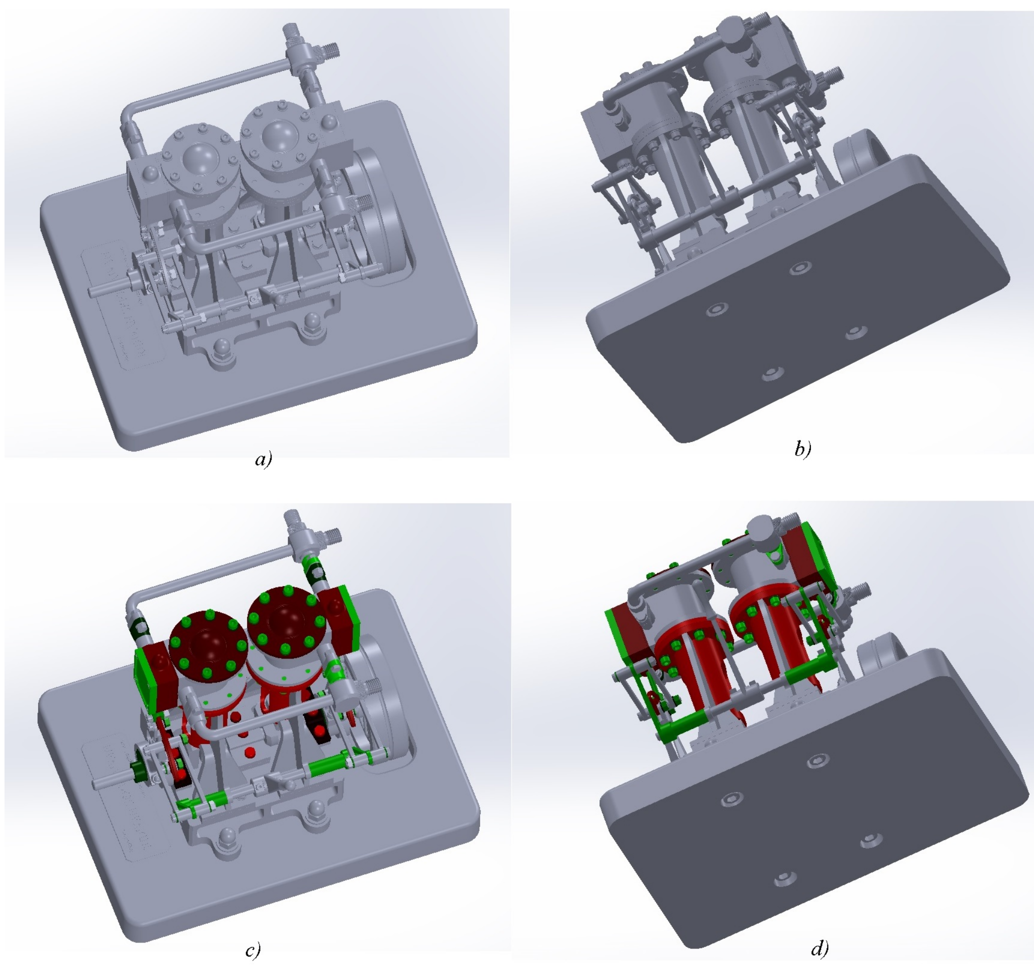


Figure 2.2: Catapult. a) Original model view 1; b) Original model view 2; c) detected patterns view 1; d) detected patterns view 2.

Figures 2.2 a) and b) show an assembly model containing 55 types of components.

From this input, the proposed algorithm detects:

- 2 circular translational patterns of length 3;
- 8 rotational patterns of length 8;
- 6 linear translational patterns of length 3;
- 4 circular translational patterns of length 3;
- 16 linear translational patterns of length 2;
- 8 rotational patterns of length 2.

Linear patterns are highlighted in red nuances and circular patterns are highlighted in green nuances in Figures 2.2 c) and d).

Chapter 3

Compound patterns

The information that the symmetric arrangement detection process may provide can be further elaborated and the information content about the object represented by the results increased. In fact, simple patterns (or patterns of first level) of repeated subparts or components, respectively in part or assembly models, can be further combined with each other aiming at detecting higher level patterns, whose repeated elements are represented by previously detected patterns.

Consider Fig. 3.1: a) represents a simple 2D situation where there are 9 REs. b) shows the result detected by the first level pattern detection, which has just been described in [1], [2] and in Chapters 1 and 2: 3 linear translational patterns are identified, each of them of length 3 and characterized by the same constant spatial distance between REs. Finally, c) represents a linear compound pattern constituted in turn by the 3 repeated first level patterns detected in the b) phase. The step represented in c) is an example of the goals of the *compound pattern detection*.

The approach proposed for compound pattern detection bases on the same fundamentals created for pattern detection of first level: a representative point for every simple pattern is computed (centroid of the pattern), then this set of points is used to

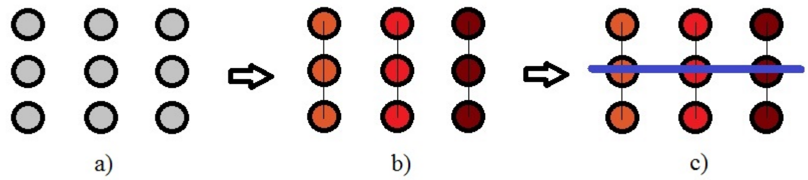


Figure 3.1: a) Starting situation; b) first level of pattern detection: simple linear translational patterns; b) second level of pattern detection: compound linear pattern.

find regular arrangements of them, aiming at finding candidate compound patterns. In fact, if a set of patterns have a regular arrangement then also their centroids do, analogously observed in the first level case. Finally, a geometric verification is provided to establish if the patterns whose centroids lie on a path are also correctly oriented.

The method has been implemented limited to composition of *linear translational patterns*, but the concept of "compound pattern" can be extended to the circular pattern cases, making some arrangements related to the different types of compound patterns that could arise.

3.1 Compound patterns in part models

In this section the approach proposed to detect compound patterns in part models is described. The method has in common several points with the first level pattern detection algorithm for part models and exploits some of its procedures.

3.1.1 Grouping surfaces for patterns

Analogously to what has been supposed for the first level case, we assume that a specific compound pattern, if it exists, has been built on a fixed face of the partial model in a determined stage of the designing process. This is why in the proposed approach compound patterns are searched within a more restricted set of search involving those patterns which are "adjacent" to faces having same lying surface. The procedure considers among the grouping surfaces GS [1], [2] considered for the simple pattern detection those adjacent to more than one simple pattern. These surfaces are referred to as *grouping surfaces for patterns* (GSFP).

Definition 3.1.1. Let $\{\mathbf{A}_0, \ldots, \mathbf{A}_{q-1}\}$ be a pattern of length q, S to be a grouping surface with list of REs LRE_S (associated to it in the previous first level stage). Then the pattern $\{\mathbf{A}_0, \ldots, \mathbf{A}_{q-1}\}$ is *adjacent* to S if $\mathbf{A}_i \in LRE_S$ for $i = 0, \ldots, q-1$. In this case S is said to be a *grouping surface for patterns* for the pattern $\{\mathbf{A}_0, \ldots, \mathbf{A}_{q-1}\}$.

For each pattern all the GSs of the first level stage are examined and if the pattern is adjacent to a GS it is added to the associated list of adjacent patterns. At the end of this process we will have a set of grouping surfaces for patterns (or GSFP) $S_0, \ldots, S_{q-1}$, $q \geq 1$, with the respectively associated $LP_0, \ldots, LP_{q-1}$ lists of the patterns, where LP_j is the list of patterns adjacent to a face with S_j as host surface.

Denoting as **S** the set of all GSFPs deriving from the pattern adjacency analysis, we order **S** by *decreasing ordering respect to the number of patterns* in the list associated to the GSFP. This is done because the highest the number of patterns on a GS is then highest is the probability to find meaningful compound symmetric arrangements. As for simple patters, GSFPs with associated a list of adjacent patterns with cardinality 1 are not considered and removed from the set of GSFPs.

3.1.2 Grouping of linear patterns

Suppose to fix a grouping surface for pattern and consider the corresponding associated list of adjacent patterns. The linear patterns identified in the first stage are classified in subsets of compatible ones, according to specific criteria.

At first all the linear translational patterns are grouped in subsets of "coherent" patterns to thin out the initial set of first level patterns, aiming at handling a smaller number of them. Formally, coherent patterns are defined as follows.

Definition 3.1.2. Let $\{\mathbf{A}_{0,0}, \ldots, \mathbf{A}_{0,n_0-1}\}, \ldots, \{\mathbf{A}_{m,0}, \ldots, \mathbf{A}_{m,n_m-1}\}$ be linear translational patterns of congruent REs and $d_0, \ldots, d_{m-1}$ the respective constant distance characterizing the pattern. We say that they are *coherent linear patterns* if $n_i = n_k$ and $d_i = d_k$ for $i, k = 0, \ldots, m-1$.

After the creation of sets of coherent linear patterns, they are ordered by decreasing ordering respect to their cardinality, such that the sets of coherent patterns with more patterns are firstly examined.

The second grouping stage is performed to support only the linear compound pattern detection (as shown by the algorithm described in Susbection 3.1.5). It consists on further subdividing each set of coherent patterns in subsets of patterns coherently oriented. The concept of "parallel" patterns is now introduced.

Definition 3.1.3. Let $\{\mathbf{A}_{i,0}, \ldots, \mathbf{A}_{i,n_i-1}\}$ and $\{\mathbf{A}_{i+1,0}, \ldots, \mathbf{A}_{i+1,n_{i+1}-1}\}$ be two linear translational patterns, $n_i, n_{i+1} > 1$. We say that the two patterns are *parallel* if the two lines associated to the two patterns are parallel lines (equal directional vector or equal up to a multiplying constant -1).

Once the subsets of coherent and parallel linear translational patterns have been identified, they are ordered by decreasing ordering respect to their cardinality and subsets containing only one element are discarded.

3.1.3 Centroid of a pattern

The representative points for patterns exploited in the path detection phase are computed as described as follows.

Suppose the set $\mathbf{C} = \{C_0, \ldots, C_n\}$ to be the set of the centroids of the REs of a linear translational pattern of length n, $n \geq 2$. Then the representative point of the pattern is computed as the centroid of the points in **C**, i.e. the point with coordinates:

$$\left(\frac{\sum_{i=0}^{n-1} x_{C_i}}{n}, \frac{\sum_{i=0}^{n-1} y_{C_i}}{n}, \frac{\sum_{i=0}^{n-1} z_{C_i}}{n} \right) \tag{3.1}$$

Such a point is called *centroid of the pattern* it refers to.

Compound patterns, as for simple patterns, arises from the preliminary results obtained by the path detection algorithm [1], [2]. The cited algorithm, together with the prior computation of the adjacency matrices, must be performed on the set of pattern centroids corresponding to a subset patterns selected under the criteria just described. The result is a set of paths of pattern centroids.

As explained in [1], [2], the possibility of handling 0-cells is more convenient than managing objects of higher dimension. The path existence analysis provides preliminary and significant information about the pattern arrangement, while lightening the following work to do on vertices, edges and faces.

3.1.4 Types of compound patterns

In this section the classes of compound patterns that the proposed approach is able to detect are described in details. We notice that the followings are only examples of the possible existing compound patterns for linear patterns and that a wide range of them can be formalized and detected, therefore the method can be further extended.

Depending on the type of path of pattern centroids has been detected, there could exist different types of compound patterns. In the following we will denote as $(C_0, \ldots, C_{q-1})$ the path of pattern centroids, as $\{\mathcal{P}_0, \ldots, \mathcal{P}_{q-1}\}$ the corresponding set of first level patterns and as $\{\mathbf{A}_{i,0}, \ldots, \mathbf{A}_{i,n_i-1}\}$ the set of REs belonging to the pattern $\mathcal{P}_i$ for $i = 0, \ldots, q-1$.

Linear translational compound pattern

Suppose $(C_0, \dots, C_{q-1})$ be a path of type line. To conclude that the set of patterns $\{\mathcal{P}_0, \dots, \mathcal{P}_{q-1}\}$ is a *linear translational compound pattern* it is necessary to examine the patterns in pairs.

Consider two patterns $\mathcal{P}_i, \mathcal{P}_{i+1}$ with consecutive pattern centroids C_i, C_{i+1} in the given path , $i \in \{0, \dots, q-2\}$. To understand if the relation between the two patterns is a translation, the the check procedure described in [1], [2] for translation relation between two repeated entities is exploited.

Let $C_i = (x_{C_i}, y_{C_i}, z_{C_i})$ and $C_{i+1} = (x_{C_{i+1}}, y_{C_{i+1}}, z_{C_{i+1}})$. Then, the candidate *translational vector* is $v = (x_v, y_v, z_v)$ computed as

$$(x_{C_{i+1}} - x_{C_i}, y_{C_{i+1}} - y_{C_i}, z_{C_{i+1}} - z_{C_i}). \tag{3.2}$$

Observe that, as $(C_0, \dots, C_{q-1})$ is a path of type line, the candidate translational vector v is the same for every pair of consecutive patterns.

To conclude that $\mathcal{P}_i, \mathcal{P}_{i+1}$ are related by translation by the candidate translational vector v, the following condition must be verified: if $\mathbf{A}_{i+1,0} = T_v(\mathbf{A}_{i,0})$ or $\mathbf{A}_{i+1,n_{i+1}-1} = T_v(\mathbf{A}_{i,0})$ then $\mathcal{P}_i, \mathcal{P}_{i+1}$ are related by translation. In formula

$$\mathcal{P}_{i+1} = T_v(\mathcal{P}_i).$$

The required condition means that one among the two extreme REs of the $(i+1)$-th pattern must be the translated of the RE at the beginning of the i-th pattern. It is sufficient comparing only the extreme REs as $\mathcal{P}_i, \mathcal{P}_{i+1}$ are both two verified parallel linear translational patterns, thus the position of the other REs are correctly arranged by translation as a consequence. This condition is valid for the other types of compound patterns as well.

Finally, we can say that the set of patterns

$$\{\mathcal{P}_0, \dots, \mathcal{P}_{q-1}\}$$

constitute a *linear translational compound pattern* if $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$ are related by a translation with the translational vector v, for $i = 0, \ldots, q-2$. In formula:

$$\mathcal{P}_{i+1} = T_v(\mathcal{P}_i) \text{ for } i = 0, \ldots, q-2.$$

Reflectional compound pattern

Let $q = 2$. To verify if two patterns $\mathcal{P}_0$ and $\mathcal{P}_1$ are related by *reflection* we exploit the procedures already built for pattern reflection [1], [2].

The candidate *reflection plane* is computed following step by step the method used in [1], [2] to compute the analogous plane for the reflectional pattern case, considering the pattern centroids $C_0 = (x_{C_0}, y_{C_0}, z_{C_0})$ and $C_1 = (x_{C_1}, y_{C_1}, z_{C_1})$ instead of the RE centroids. We denote the plane obtained as $\mathcal{P}_r$.

To conclude that $\mathcal{P}_0, \mathcal{P}_1$ are related by reflection by the candidate reflection plane $\mathcal{P}_r$, the following condition must be verified: if $\mathbf{A}_{1,0} = R_\pi(\mathbf{A}_{0,0})$ or $\mathbf{A}_{1,n_1-1} = R_\pi(\mathbf{A}_{0,0})$ then $\mathcal{P}_0, \mathcal{P}_1$ are related by reflection. In formula

$$\mathcal{P}_1 = R_\pi(\mathcal{P}_0).$$

The verify is limited to the extreme REs analogously to the translational case. In case of positive answer $\{\mathcal{P}_0, \mathcal{P}_1\}$ constitutes a *reflectional compound pattern*.

Circular translational compound pattern

Suppose $(C_0, \ldots, C_{q-1})$ be a path of type circumference. To conclude that the set of patterns

$\{\mathcal{P}_0, \ldots, \mathcal{P}_{q-1}\}$ is a *circular translational compound pattern* we examine the patterns in pairs as in the linear compound case.

We consider for a while two patterns $\mathcal{P}_i, \mathcal{P}_{i+1}$ with consecutive pattern centroids C_i, C_{i+1} in the given path , $i \in \{0, \ldots, q-2\}$.

The difference respect to the linear situation is that the candidate translational vector changes direction for each pair of consecutive patterns, while the Euclidean norm of the vector is a constant value (analogously observed for the circular translational case for first level patterns [1], [2]). The Euclidean norm of the translational vectors is the repeated distance between centroids involved in the path of type circumference.

To conclude that the set of patterns $\{\mathcal{P}_0, \ldots, \mathcal{P}_{q-1}\}$ constitute a *circular translational compound pattern* we must verify that for $i = 0, \ldots, q-2$ the patterns $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$ are related by translational function with the translational vector v_i, where v_i is computed as $(x_{C_{i+1}} - x_{C_i}, y_{C_{i+1}} - y_{C_i}, z_{C_{i+1}}) - z_{C_i}$. In formula:

$$\mathcal{P}_{i+1} = T_{v_i}(\mathcal{P}_i) \text{ for } i = 0, \ldots, q-2.$$

Rotational compound pattern

Suppose $(C_0, \ldots, C_{q-1})$ be a path of type circumference. We examine which are the condition to say that the set of patterns $\{\mathcal{P}_0, \ldots, \mathcal{P}_{q-1}\}$ is a *rotational compound pattern*.

As in the previous situations, we consider two patterns $\mathcal{P}_i, \mathcal{P}_{i+1}$ with consecutive pattern centroids C_i, C_{i+1} in the given path , $i \in \{0, \ldots, q-2\}$.

The candidate *rotation matrix* is computed following the procedure described in Subsection 1.1.2 for the simple rotational case, considering the current pattern centroids instead of the RE centroids. Let the candidate rotation matrix to be $R_{r,\theta}$.

To conclude that $\mathcal{P}_i, \mathcal{P}_{i+1}$ are related by rotation by the candidate rotation matrix $R_{r,\theta}$ the following condition must be verified: if $\mathbf{A}_{i+1,0} = R_{r,\theta}(\mathbf{A}_{i,0})$ or $\mathbf{A}_{i+1,n_{i+1}-1} = R_{r,\theta}(\mathbf{A}_{i,0})$ then $\mathcal{P}_i, \mathcal{P}_{i+1}$ are related by rotation (the description of the steps to verify for first level rotation is in Subsection 1.1.2). In formula

$$\mathcal{P}_{i+1} = R_{r,\theta}(\mathcal{P}_i).$$

Finally, we say that the set of patterns

$$\{\mathcal{P}_0, \ldots, \mathcal{P}_{q-1}\}$$

constitute a *rotational compound pattern* if $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$ are related by rotation with the rotation matrix $R_{r,\theta}$, for $i = 0, \ldots, q-2$. In formula:

$$\mathcal{P}_{i+1} = R_{r,\theta}(\mathcal{P}_i) \text{ for } i = 0, \ldots, q-2.$$

3.1.5 Compound pattern detection algorithm for part models

In this subsection we illustrate the sequence of steps of the *compound pattern detection algorithm for part models*. Also in this algorithm the *path detection algorithm*, already introduced in [1], [2], is recalled in the algorithm with the name "*FIND_ALL_PATHS*" constituting again a fundamental part of the process.

Here follows the pseudo-code that summarizes the flow of operations performed for the compound pattern detection.

COMPOUND_PATTERN_DETECTION_FOR_PART

Input: list, *list of patterns*;
Compound_pattern_list, *list of the compound patterns to be updated*;
Compound_pattern_length2_list, *list of the compound patterns of length 2 to be updated*.

Output: Compound_pattern_list;
Compound_pattern_length2_list.

List_adjacency_matrices = *COMPUTE_ADJACENCY_MATRICES (*list*)*
THIN_OUT_AND_ORDERING(List_adjacency_matrices)
for each *(*$\mathcal{M}$ *in* List_adjacency_matrices*)* ***do***
    List_of_paths = *FIND_ALL_PATHS* ($\mathcal{M}$)

ORDERING(List_of_paths)

for each *($\mathcal{P}$ in* List_of_paths*)* ***do***

\\ *Verification of the identified paths*

VERIFY_COMPOUND_PATTERNS_AND_UPDATE_LISTS(

$\mathcal{P}$, Compound_pattern_list, Compound_pattern_length2_list

UPDATE_OTHER_DATA(

List_of_paths,List_adjacency_matrices, Compound_pattern_length2_list

Remove $\mathcal{P}$ *from* List_of_paths

end for

Remove $\mathcal{M}$ *from* List_adjacency_matrices

end for

return Compound_pattern_list, Compound_pattern_length2_list

MAIN_COMPOUND_PATTERN_DETECTION_FOR_PART

Input: Pattern_list, *list of first level patterns*

GS_list, *list of first level grouping surfaces*

Output: Compound_pattern_list, *list of compound patterns of REs*.

Compound_pattern_length2_list, *list of compound patterns length* 2 *of REs*

Compound_pattern_list = *empty*

Compound_pattern_length2_list = *empty*

List_Of_GSforPatterns = *BUILD_LIST_OF_GS_FOR_PATTERNS(*Pattern_list, GS_list*)*

for each *(*GSforPatterns *in* List_Of_GSforPatterns*)* ***do***

List_of_linear_patterns = *SELECT_LINEAR_FROM_GS_FOR_PATTERNS(*GSforPatterns*)*

List_of_list_of_coherent_patterns = *FIND_COHERENT_PATTERNS(*List_of_linear_patterns*)*

for each *($\mathcal{L}$ in* List_of_lists_of_coherent_patterns*)* ***do***

List_of_lists_of_parallel_patterns = *FIND_PARALLEL_PATTERNS($\mathcal{L}$)*

```
if ℒ.count = 2 then
    bool = VERIFY_IF_IS_REFLECTIONAL_PATTERN(ℒ, Compound_pattern_length2_list)
    if bool=false & List_of_lists_of_parallel_patterns.count = 1 then
        bool = VERIFY_IF_IS_TRANSLATIONAL_PATTERN(ℒ,
            Compound_pattern_length2_list)
    end if
else
    for each (ℳ in List_of_lists_of_parallel_patterns) do
        if ℳ.count = 2 then
            bool = VERIFY_IF_IS_REFLECTIONAL_PATTERN(ℳ,
                Compound_pattern_length2_list)
            if bool=false then
                bool = VERIFY_IF_IS_TRANSLATIONAL_PATTERN(ℒ,
                    Compound_pattern_length2_list)
            end if
        else
            COMPOUND_PATTERN_DETECTION_FOR_PART(ℳ,
                Compound_pattern_list, Compound_pattern_length2_list)
        end if
        Remove ℳ from List_of_lists_of_parallel_patterns
    end for
    REMOVE_PATTERNS_ALREADY_SET_IN_COMPOUND_PATTERN(ℒ,
        Compound_pattern_list)
    if ℒ.count = 2 then
        bool = VERIFY_IF_IS_REFLECTIONAL_PATTERN(ℒ,
            Compound_pattern_length2_list)
    else
```

\\ *Search of rotational compound patterns*

COMPOUND_PATTERN_DETECTION_FOR_PART*($\mathcal{L}$,*

Compound_pattern_list, Compound_pattern_length2_list*)*

end if

end if

end for

end for

\\ *Final attempt to find patterns of length 2*

ARRANGEMENTS_IN_PATTERNS_OF_LENGTH2 (

Compound_pattern_length2_list, Compound_pattern_list, List_Of_GSforPatterns)

return Compound_pattern_list, Compound_pattern_length2_list

In the following we provide a description of the main steps of the proposed approach for the detection of compound patterns of REs, considering as a starting point the knowledge of the first level patterns.

1. *Building of GSFPs*. The list of grouping surfaces for patterns is created and updated by elaborating the information already available about GSs from the first level analysis. For each pattern we select the set of GSs such that every RE belonging to the pattern is adjacent to it (see Definition 3.1.1). Once built this set, the following points are repeated for each GS in the set of GSFPs.

2. *Grouping of patterns*. We select only the linear patterns from the set of patterns associated to a GSFP. Then, the linear patterns are subdivided in subsets, each of them containing only coherent patterns (see Def. 3.1.2). These subsets are ordered in decreasing ordering respect their cardinality. The final grouping stage consists of furtherly subdividing each set of coherent pattern in subsets of parallel linear patterns, coherently to the Definition 3.1.3, and then the set of coherent par-

allel patterns is ordered in decreasing ordering respect to their cardinality again. Each subsets of linear patterns containing only 1 element in deleted from the list.

3. *Parallel pattern analysis*. If a set of coherent parallel patterns has cardinality equal to 2 it is a candidate reflectional or translational pattern, so if the first verify fails the second one is performed. If the set of coherent parallel patterns has cardinality greater than 2, then the adjacency matrices are computed on the set of pattern centroids, the path detection algorithm is applied and the resulting paths verified, following the same reasonings and choices used in [1], [2]. The patterns whose centroids lie on a linear path and that overcome the corresponding geometric analysis are so classified as linear translational compound patterns. On the other hand, the patterns whose centroids lie on a circular path and that verify the geometric analysis are classified as circular translational compound patterns.

4. *Non-parallel pattern analysis*. After all subsets of coherent parallel patterns of the current GSFP have been analysed, the rotational patterns are searched in the set of first level patterns that have not been arranged in a compound pattern yet. The set of remaining patterns is built by removing the already arranged patterns from the current set of coherent patterns. If the resulting set contains only 2 patterns it is a candidate reflectional and the corresponding verification is performed. If the resulting set of patterns has cardinality greater than 2, then the adjacency matrices computation and the and the path detection algorithm occur by considering the pattern centroids as reference points. The patterns whose centroids lie on a circular path and that verify the geometric analysis are classified as rotational compound patterns.

5. *Final attempt to find patterns of length* 2. At the end of the analysis of the GSs for patterns there could remain a set of first level patterns not yet in a detected compound arrangements of length at least 3. For these patterns we decided to

associate them, if it exists, to an already found symmetric arrangement of length 2 containing them (these patterns are found "by chance" while we were looking for longer patterns, as we will explain below). If such a pattern of length 2 does not exist, a possibility (not implemented yet) is to try to couple the remaining pattens belonging to the same GSFP's list two by two, verifying if they are related by a regular arrangements (translation, reflection), starting the coupling attempting from the nearest couples of patterns.

3.2 Compound patterns in assembly models

The set of procedures defined for the detection of compound patterns of REs in part models can be easily extended to the assembly case. As just mentioned in Chapter 2, in assembly models the repeated elements are constituted by the components and after the first level analysis the result is a set of simple patterns of components. Here grouping surfaces are not a considered element in the process, either in the simple and compound pattern detection approach for assembly models, and so the operations referred to the analysis of the set GSs are not performed. Anyway the followed general reasoning is almost the same and the considered types of compound patterns are analogous (except for the reflectional type, which is not applicable to the assembly case, as just observed in 2.2.1):

- *linear translational compound pattern;*
- *circular translational compound pattern;*
- *rotational compound pattern.*

The steps of the approach are briefly described here. For the computation of the adjacency matrices and the application of the path detection algorithm it is sufficient to consider as representative point for a first level pattern the *centroid of the origins* of the repeated components. Then, most of the procedures adopted for the detection of the first level patterns (Section 2.2) are applied. In particular:

- the translation procedures (see Subsec. 2.2.1) are used for the detection of linear and circular translational compound patterns;

- the rotational procedures (see Subsec. 2.2.3) are applied for the detection of rotational compound patterns.

3.3 Results of the compound pattern methods

We report some of the results obtained applying the developed functions for the compound pattern detection, both on part and assembly models. As usual, the functions are developed in C$\sharp$ and integrated as a plug-in in the SolidWorks$^{©}$ CAD system.

In the following examples the first level pattern detection algorithm is applied, then the procedures referred to the compound pattern detection are performed on the results of the first stage.

Some of the examples have been already used in [1] and [2] for the tests referred to the first level phase. Here we add to the first level information new data about the existence of compound patterns.

3.3.1 Electrical component

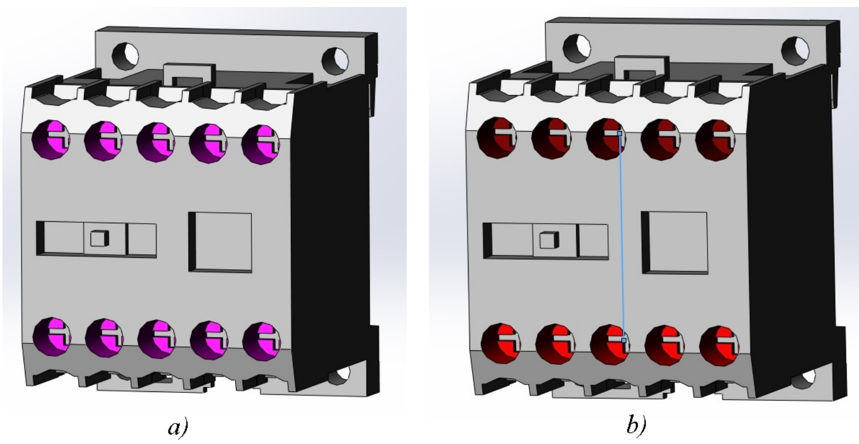
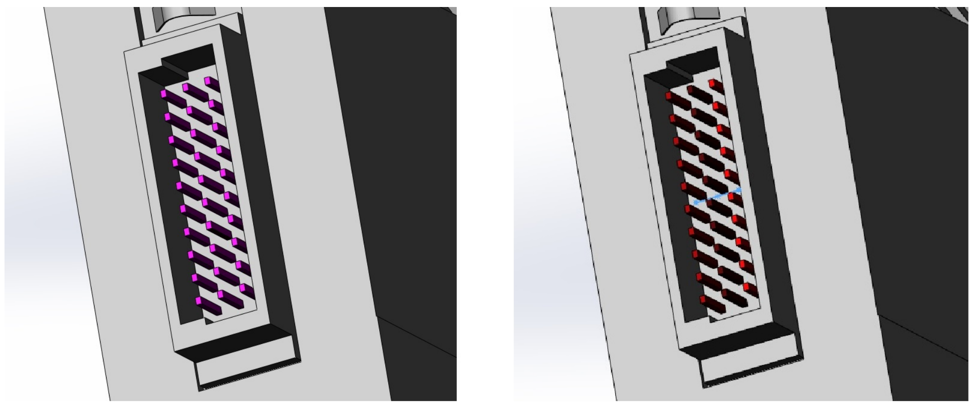


Figure 3.2: Electrical component. a) Selected REs; b) detected first level patterns and compound pattern.

We consider the part model in Figure 3.2 a), where the faces corresponding to the input REs are highlighted in violet. Each RE is composed by 14 planar faces and 1 cylindrical face.

In this situation the algorithm detects 2 translational patterns, each of them constituted by 5 of the initially selected REs, each of them coloured in a red nuance (Figure 2.1 b). Moreover, the blue line represents a linear translational compound pattern of length 2 made of the 2 first level patterns.

3.3.2 Pins



a) *b)*

Figure 3.3: Pins. a) Selected REs; b) detected first level patterns and compound pattern.

In Figure 3.3 a) we highlighted the faces corresponding to the input REs in violet. In the current part model each RE is composed by 4 planar faces. The input is constituted by 30 REs.

The algorithm detects 3 translational patterns, each of them constituted by 10 of the initially selected REs, each of them coloured in a red nuance (Figure 3.3 b). The blue line represents a linear translational compound pattern of length 3 made of the 3 first level patterns.

3.3.3 Castle

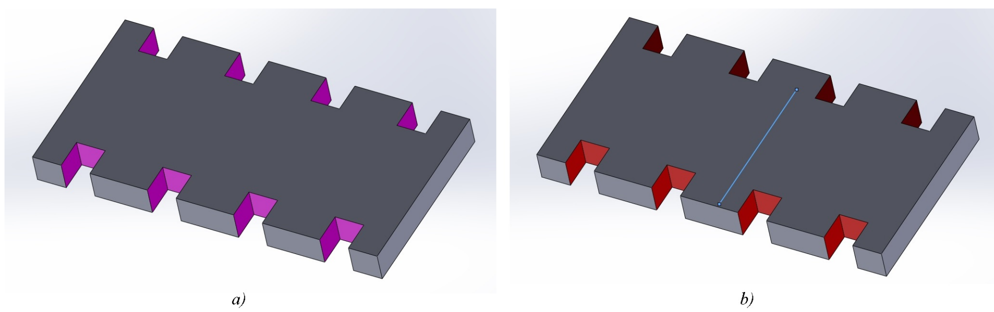
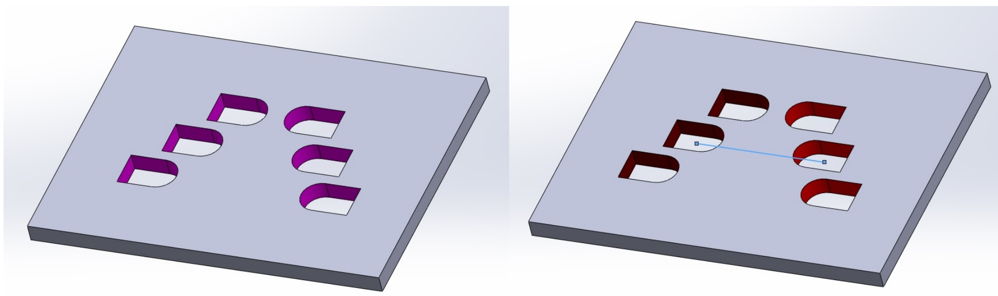


Figure 3.4: Castle component. a) Selected REs; b) detected first level patterns and compound pattern.

Consider the part model in Figure 3.4 a). The faces corresponding to the input REs are highlighted in violet. The input is constituted by 8 REs and each RE is composed by 3 planar faces.

In this situation the algorithm detects 2 translational patterns, each of them constituted by 4 of the initially selected REs, in red nuances (Figure 2.1 b). As for compound patterns, the blue line represents a reflectional compound pattern of length 2 made of the 2 first level patterns.

3.3.4 Reflectional holes



a) b)

Figure 3.5: Reflectional holes. a) Selected REs; b) detected first level patterns and compound pattern.

In Figure 3.5 a) we highlighted the faces corresponding to the input REs in violet. In the current part model each RE is composed by 3 planar faces and 1 cylindrical face. The input is constituted by 6 REs.

The algorithm detects 2 translational patterns, each of them constituted by 3 of the initially selected REs, each of them coloured in a red nuance (Figure 3.5 b). The blue line represents a reflectional compound pattern of length 2 made of the 2 first level patterns.

3.3.5 Circular translational holes

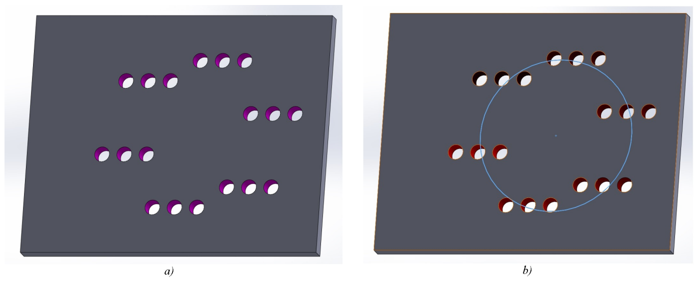


Figure 3.6: Circular translational holes model. a) Selected REs; b) detected first level patterns and compound pattern.

Consider the part model in Figure 3.6 a). The input is constituted by 18 circular holes and each hole (RE in violet) is composed by 1 cylindrical faces.

In this situation the algorithm detects 6 linear translational patterns, each of them constituted by 3 of the initially selected REs, in red nuances (Figure 3.6 b). As for compound patterns, the blue circle represents a circular translational compound pattern of length 6 made of the 6 first level linear patterns.

3.3.6 Circular plate

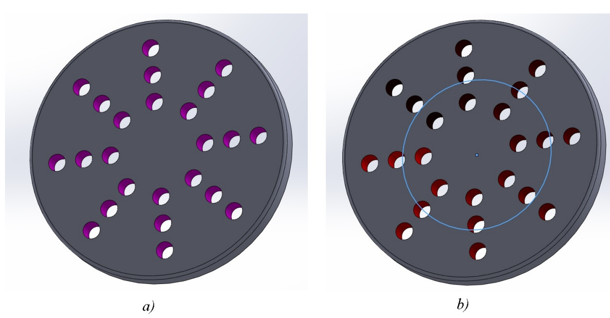


Figure 3.7: Circular plate. a) Selected REs; b) detected first level patterns and compound pattern.

In Figure 3.7 a) we highlighted the faces corresponding to the input REs in violet. In the current part model each RE is composed by 1 circular hole (1 cylindrical face). The input is constituted by 24 REs.

The algorithm detects 4 linear translational patterns, each of them constituted by 3 of the initially selected holes, coloured in red nuances (Figure 3.7 b). The blue circle represents a rotational compound pattern of length 8 made of the 8 first level patterns.

3.3.7 Plate with multiple patterns

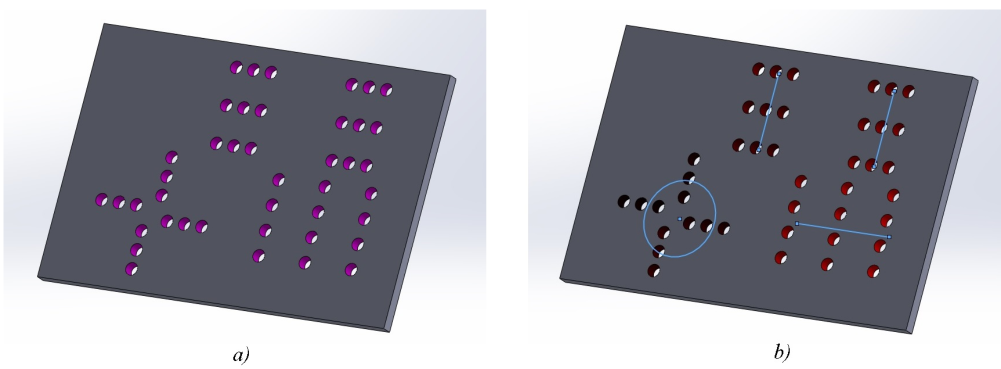


Figure 3.8: Plate with multiple patterns model. a) Selected REs; b) detected first level patterns and compound pattern.

In Figure 3.8 a) we highlighted the faces corresponding to the input REs (circular holes) in violet. In the current part model each RE is composed by 1 cylindrical face. The input is constituted by 42 REs.

The algorithm detects 13 linear translational patterns: 10 of them are constituted by 3 of the initially selected REs, the other 3 are made of 4 circular holes. The patterns are coloured in red nuances (Figure 3.8 b). The compound pattern analysis detects 3 linear translational compound patterns, each of them constituted by 3 first level patterns and 1 rotational pattern of length 4 (see the blue lines).

3.3.8 Plate with extrusions 1 (assembly model)

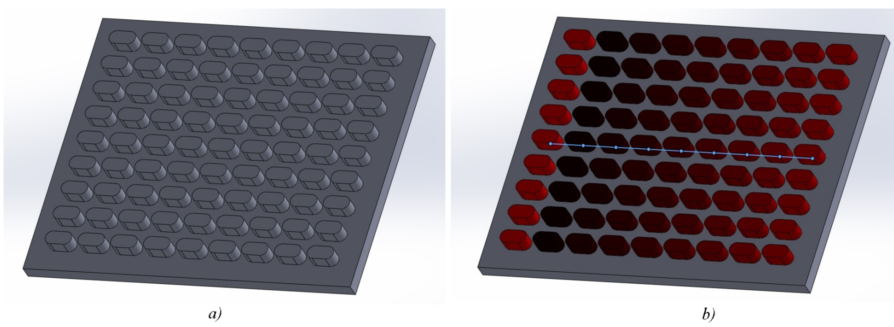


Figure 3.9: Plate with extrusions 1. a) Original model; b) detected first level patterns and compound pattern.

In Figure 3.9 a) there is represented an assembly model containing 2 different types of components: the plate and 81 extrusions.

The algorithm detects 9 linear translational patterns, each of them constituted by 9 extrusions, coloured in red nuances (Figure 3.9 b). The blue line represents a linear translational compound pattern of length 9 made of the 9 first level patterns of extrusions.

3.3.9 Plate with extrusions 2 (assembly model)

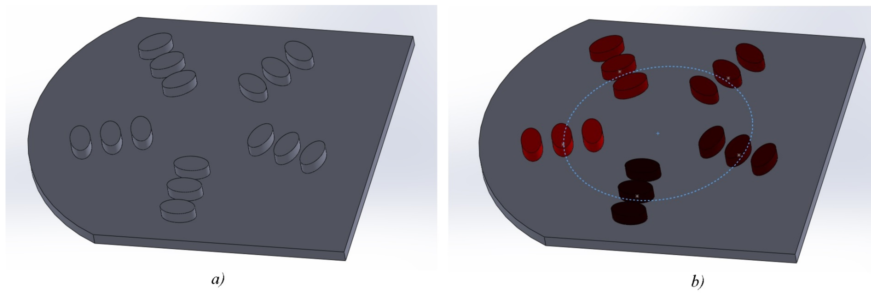


Figure 3.10: Plate with extrusions 2. a) Original model; b) detected first level patterns and compound pattern.

Observe Figure 3.10 a): the assembly model contains 2 different types of components which are the plate and 15 elliptical extrusions.

The algorithm detects 5 linear translational patterns, each of them constituted by 3 extrusions, coloured in red nuances (Figure 3.10 b). The blue circle represents a rotational compound pattern of length 5 made of the 5 first level patterns of extrusions.

[page left intentionally blank]

Bibliography

[1] L. Chiang. *Identification of Patterns of Repeated Parts in Solid Objects*. Master Thesis. Academic year 2013/2014.

[2] L. Chiang., F. Giannini, M. Monti *Identification of Patterns of Repeated Parts in Solid Objects*. Technical report Nr. 16-14, Institute of Applied Mathematics and Information Technologies “E. Magenes” (CNR), Genoa. 2016.

[3] L. Zaccarian. *Dispense del corso di Robotica con Laboratorio*. Course notes. 2002. http://control.disp.uniroma2.it/zack/LabRob/dispense.pdf (visited on 21/07/2015)

Recent titles from the IMATI-REPORT Series:

16-01: *Optimal strategies for a time-dependent harvesting problem*, G.M. Coclite, M. Garavello, L.V. Spinolo.

16-02: *A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0,* R. Vázquez.

16-03: *Defect detection in nanostructures*, D. Carrera, F. Manganini, G. Boracchi, E. Lanzarone.

16-04: *A study of the state of the art of process planning for additive manufacturing*, M. Livesu, M. Attene, M. Spagnuolo, B. Falcidieno.

16-05: *Calcolo di descrittori ibridi geometria-colore per l'analisi di similarità di forme 3D,* A. Raffo, S. Biasotti.

16-06: *An appointment scheduling framework to balance the production of blood bags from donation*, S. Baş, G. Carello, E. Lanzarone, S. Yalçındağ.

16-07: *From lesson learned to the refactoring of the DRIHM science gateway for hydro-meteorological research,* D. D'Agostino, E. Danovaro, A. Clematis, L. Roverelli, G. Zereik*,* A. Galizia.

16-08: *Algorithms for the implementation of adaptive isogeometric methods using hierarchical splines*, E.M. Garau, R. Vázquez.

16-09: *Feature curve identification in archaeological fragments using an extension of the Hough transform,* M.L. Torrente, S. Biasotti, B. Falcidieno.

16-10: *Comparing methods for the approximation of rainfall fields in environmental applications,* G. Patané*,* A. Cerri, V. Skytt, S. Pittaluga, S. Biasotti, D. Sobrero, T. Dokken, M. Spagnuolo.

16-11: *Persistence-based tracking of rainfall field maxima*, S. Biasotti, A.Cerri, S. Pittaluga, D. Sobrero, M. Spagnuolo.

16-12: *Studio e sviluppo di componenti semantiche riusabili per la progettazione di serious game. Applicazioni all'area edutainment,* A. Repetto, C.E. Catalano.

16-13: *Optimal-order isogeometric collocation at Galerkin superconvergent points,* M. Montardini, G. Sangalli, L. Tamellini.

16-14: *Identification of patterns of repeated parts in solid objects*, L. Chiang, F. Giannini, M. Monti.

16-15: *Identification of patterns of repeated parts in solid objects Part II*, L. Chiang, F. Giannini, M. Monti.

Istituto di Matematica Applicata e Tecnologie Informatiche ``*Enrico Magenes",* CNR
Via Ferrata 5/a, 27100, Pavia, Italy
Genova Section: Via dei Marini, 6, 16149 Genova, Italy • **Milano Section:** Via E. Bassini, 15, 20133 Milano, Italy

http://www.imati.cnr.it/