

Identification of Regularities in CAD Part and Assembly Models

L. Chiang, F. Giannini, and M. Monti^(✉)

Institute of Applied Mathematics and Information Technology
“E. Magenes” – CNR, Genoa, Italy
lisa.chiang.sv@gmail.com,
{giannini, monti}@ge.imati.cnr.it

Abstract. The identification of regular patterns of congruent features in CAD models can enrich the object representation by a set of higher level information, which can be exploited for the reuse of the part model. In this paper, a method for the detection of regular patterns and symmetries of repeated subparts in B-Rep part and assembly models is proposed. The method is implemented as a plug-in of a commercial CAD system and detects linear, circular translational, rotational and reflectional patterns of congruent sub-parts of the model.

Keywords: Regular patterns · Symmetries in CAD models · Design intent

1 Introduction

In the design of manufactured parts, symmetry has been gaining increasing interest because of economical, manufacturing, functional, or aesthetic considerations [1]. In mathematics, the term “symmetry” refers to a function that, once applied to a shape, leaves it unchanged. In computer graphics and engineering, the meaning is extended to a wider concept, including not only the classical geometric property referred to a single shape, but also a “regularity” intended as arrangements of repeated sub-parts of the model subjected to geometric transformations as reflections, translations, rotations or combinations thereof [2].

The most popular models employed by CAD (Computer-Aided Design) systems adopted by designers in industrial field are boundary representation (B-Rep) models. It has been estimated that around 80 % of all design tasks concerns the adaptation of existing design models to new requirements [3]. In this context, recognizing intentional patterns of congruent features adopted by CAD designers, such as circular, rectangular, or even user-defined patterns can enrich the object representation with a set of higher level information, which can be further exploited either in the object production or for the creation of new parts as a variation of the existing ones. For instance, changes on the characteristics of the identified patterns can automatically change the position of the related elements, e.g. modifying the radius of a circular pattern of holes automatically modifies the positions of the associated holes. Moreover, once added this set of

information, the representation of a solid can be further compressed by exploiting the symmetry information in an optimal way. For example, an object characterized by a global reflectional symmetry can be stored by keeping only data related to half an object and then annotating that the entire object can be obtained by replicating its half part with a reflection in the specified plane. Similarly, for repeated elements arranged according to a predefined pattern, only a representative shape element can be stored together with the arrangement rule.

During the product development process, symmetry properties of mechanical components can be used to compute tool path trajectories of a machining process and to structure these trajectories in order to optimize the tool displacements. In practice, this helps locating the machining and assembly arms when the product needs to be machined and assembled [5]. Similarly, in the shape adaptation process for finite-element analysis symmetries can be used to facilitate the simplification of the model for the validation stage [4]. In some cases, B-Rep models arise from reverse engineering process thus the design intent is completely lost and detecting symmetries could be very helpful to recover it [6].

Finally, symmetry properties may facilitate also the retrieval of CAD models [2]; considering queries that include symmetry constraints, the searching can be more effective and selective than choosing more general and less significant searching parameters, such as the number and types of faces.

In this paper a method for the detection of regular patterns and symmetries of repeated subparts in B-Rep part and assembly models is proposed. After an overview of the related works, in Sect. 3 the proposed approach is illustrated, while Sect. 4 provides some of the achieved results. Section 5 concludes the paper.

2 Related Works

The concept of symmetry has received significant attention in computer graphics and computer vision research in recent years. Numerous methods have been proposed to find and extract geometric symmetries and exploit such high-level structural information for a wide variety of geometry processing tasks. In [2] the main existing symmetry detection algorithms are classified by considering if the resulting symmetries involve the whole input object or only parts of it. The first class of algorithms generally exploit an important property shared by all models exhibiting a global symmetry: the planes of reflection and/or the axes of rotation pass through their center of mass of the object. This property greatly reduces the search field for symmetry extraction.

Among the research works more targeted to engineering applications, the algorithms aimed at partial symmetry detection [1, 4–7] share many similarities in their structure and the main stages of these approaches can be summarized as follows:

- Feature/sub-part selection to decompose the entire model in a set of smaller subsets of interest for the computation.
- Identification of local symmetries information.
- Identification of the meaningful partial symmetries from the collection of the detected local symmetries.

We propose a method that analogously to [6, 7] detects regular arrangements of replicated object sub-parts. In our case, similarly to [7], we exploit specific characteristic points to make easier the pattern identification. Differently from the symmetry recognition approaches adopted in image or mesh processing [2] which mostly work on dense set of points, here analogously to [6] we aim at identifying all the possible regularities in a limited set of points which are recognized to be meaningful for the model.

Differently from [6] the proposed method does not need any a priori knowledge concerning the correspondence between the B-Rep elements of the repeated entities (i.e. which vertices of a part respectively correspond to vertices of one other). It requires as input a set of faces of a B-Rep model representing the repeated parts or sub-parts (RE), either given by the user or automatically identified by a dedicated application. Moreover, it can also be applied to discover regular configurations of components and subassemblies in an assembly model where the repeated parts are explicitly available.

3 The Proposed Approach

The method here proposed discovers regular configurations in a set of REs by applying a series of grouping and filtering processes to reduce the complexity and the number of elements on which to perform the symmetry rule detection. The application of the first filter is founded on the consideration that if the REs are positioned in the object according to a specific rule, they have been inserted at some design step on a single face, which may be split and modified at successive modelling steps. This brings to the notion of grouping surfaces that will be described below. Another important characteristic of the types of arrangements we are considering is the constant distance among the REs that allows dividing them in compatible sub-groups. The computation of the distances and the detection of the type of pattern are simplified by the consideration that if a set of congruent sub-parts is characterized by a regular arrangement, then also the respective centroids do. Vertices of the model and other characteristic points cover a fundamental role in the symmetry detection. Firstly, these points are used to compute the centroid for every RE in the set provided as input. Secondly, once a regular arrangement of centroids is identified, they are exploited to verify if the corresponding REs constitute a regular arrangement as well.

The centroid C_j of the set of n vertices V_j of the RE A_j is the point with coordinates defined by:

$$C_j = \left(\frac{\sum_{i=0}^{n-1} x_{V_{j,i}}}{n}, \frac{\sum_{i=0}^{n-1} y_{V_{j,i}}}{n}, \frac{\sum_{i=0}^{n-1} z_{V_{j,i}}}{n} \right)$$

To transpose the problem of finding a regular configuration of sub-parts to a problem of finding a regular configuration of centroid points, it is necessary that these points are computed in the same way for every RE provided by the initial set. In other words, it means that if we could overlap two REs (which are congruent) their corresponding computed representative points should overlap too. The risk of non-homogeneous way of computation of centroids in the various REs rises when the REs faces are not maximized or contain closed curved edges. Therefore, to get rid of possible noise derived from different object creation processes, faces are first maximized by comparing the underlying surfaces of the adjacent faces. Then, since closed edges in the B-Rep do not have any vertex, two new vertices are symmetrically inserted at the initial and mid curve positions of each closed curved edge of the REs. Moreover, since only considering the B-Rep vertices gives rise to ambiguous centroids when curved edges are present, as they cannot discriminate the position of concave and convex curved edges, for each curved edge an additional point is added at the middle of the edge. In the example of Fig. 1 the red points indicate fictitious vertices added for a non-ambiguous centroid computation.

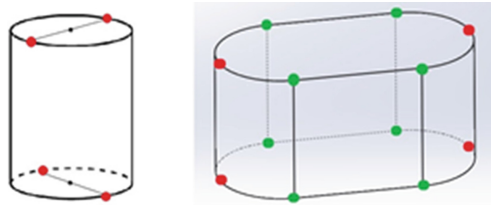


Fig. 1. Examples of points considered for the centroid computation: in green B-Rep vertices, in red the added points (Color figure online).

In the following, we introduce some concepts that are widely used in the adopted method.

Grouping surfaces

As earlier said, we started from the assumption that if a regular arrangement of repeated subparts corresponds to a specific design intent, it was likely created at a given design stage on a unique face, which may be split or modified at successive modelling steps. Therefore, we decided to decompose the original set of REs into subsets sharing the same adjacent surface: if a face of the RE A_i is adjacent to the face F of the B-Rep and the surface of F is S , we associate the RE A_i to the surface S . S is called grouping

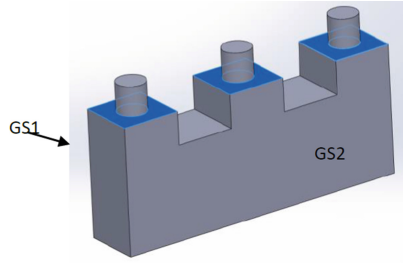


Fig. 2. Examples of REs associated with several grouping surfaces (the 2slots) and of grouping surface hosting several B-Rep faces (the blue ones) (Color figure online).

surface (GS). In this way, we find a set of GSs and each of them is associated to a set of REs. This RE set decomposition is not a partition, since the intersection of two sets of REs associated to two different GSs can be not empty, as a RE A_i could be adjacent to different faces with different host surfaces, associating it to more than one GS, as in the example in Fig. 2.

In the depicted example, the three repeated bosses have only the blue GS associated, which is also associated, together with GS1 and GS2 to the two slots. GSs associated to only one RE are not considered.

Adjacency matrices at constant distance

We focused on symmetric regularities characterized by a constant distance between two centroids of two consecutive REs; thus, for each grouping surface, we group together all the REs whose centroid are at a constant distance d . We define a d -adjacency matrix as follows: Let $\{C_0, \dots, C_{n-1}\}$ be a set of points in \mathbb{R}^3 and d a real number, $d > 0$, we call d -adjacency matrix the $n \times n$ symmetric matrix M_d such that:

$$M_d(i,j) = \begin{cases} 1 & \text{if } d(C_i, C_j) = d \\ 0 & \text{if } d(C_i, C_j) \neq d \end{cases}$$

Where $M_d(i,j)$ denotes the entry of M_d at position (i, j) , $\forall i, j = 0, \dots, n-1$.

A d -adjacency matrix can be viewed as a network of points in \mathbb{R}^3 each of them connected to one or more points of the network by a straight arc of length d .

For every GS found, a list of adjacency matrices at constant distance d is then created, one for each distance d found between the centroids of the REs associated to GS. The adjacency matrices are used to quickly identify the sets of equidistant centroids and then the possible patterns involving as many as possible REs.

The whole algorithm may be summarized as follows:

```

Program_Find_Regular_Pattern(RE_List)
{For any RE find centroid coordinates and possible
corresponding grouping surfaces}
For each (RE in RE_list) do
  {Add necessary vertices }
  Vertices(RE) = Add_Fictitious_Vertices(RE);
  {Calculate the centroid of the set of vertices of RE}
  C(RE) = Centroid_Computation(Vertices(RE));
  {Get the possible grouping surfaces for the RE and add
  them to the GS list if not already present}
  GS = Get_GS(RE);
  GS_list = Add_element_to_list(GS_list, GS)
  list_of_RE_grouped_by_GS(GS) = Add_element_to_list(RE)
end for
{All GSs associated to a REs list containing at least two
elements are ordered by descending order criteria respect to
the number of elements in the associated list of REs}
Thin_Out_And_Ordering(GS_list);
{For any GS search for pattern among the grouped REs }
For each (GS in GS_list) do
  If list_of_RE_grouped_by_GS(GS) > 2 then
    {for any distance  $d$  among the REs create the adjacency matrix
    and add it to the related list, which is then ordered
    according the increasing value of  $d$  }
    List_adjacency_matrices = Compute_Adjacency_Matrices(GS);
  Thin_Out_And_Ordering(List_adjacency_matrices);
  For each M in List_adjacency_matrices do
    List_of_paths = Path_detection_algorithm (M);
    For each P in List_of_paths do
      Pattern_Assessment (P);
    End for
  End for
End for

```

3.1 Path Detection Algorithm

The algorithm aims at identifying all the possible paths in the centroid network represented by a given d -adjacency matrix. These sequences of centroids are made of at least three centroids satisfying specific geometric conditions. In the developed method, we focused on symmetric arrangements of REs whose centroids lie all on a line or on a circumference.

Let $C = \{C_0, \dots, C_{n-1}\}$ be a set of centroids of REs associated to the same grouping surface; we call path of length l with $3 \leq l \leq n$, an ordered sequence of l centroids (B_0, \dots, B_{l-1}) with $B_j \in C$ for $j = 0, \dots, l-1$ such that all lie on the same line (path of type “linear”) or all lie on the same circumference (path of type “circular”) and such that $d(B_i, B_{i+1}) = d$ with $d > 0$ and for $i = 0, \dots, l-2$.

A path is built step by step, by first choosing an initial seed path of three centroids (*seed1*, *seed2*, *seed3*) and, once the type of the path that is going to be built has been established, by adding every time a new centroid to the current path if possible.

If the three initial points are aligned it will be a path of type linear, otherwise, it will be a path of type circular. In both cases, at first the attempt of expansion is done in the “*seed1* to *seed2*” direction; when the expansion in this direction is no longer possible then a second expansion attempt is done in the “*seed2* to *seed1*” direction. Seed points are chosen starting from the so-called multi-branch points, i.e. points equidistant from more than 2 points, e.g. points 1, 2, 3, 11 in Fig. 3.

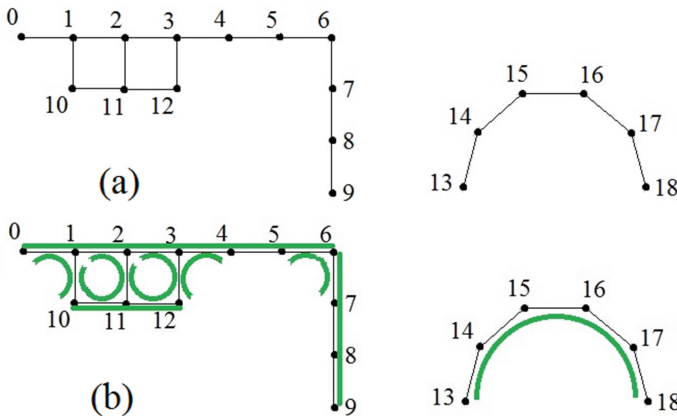


Fig. 3. Example of detected paths in a set of centroids of REs on the same grouping surface and at a constant distance.

Let us discuss when a new point can be added to the current path and thus when the expansion is possible. Suppose the current path in expansion to be (B_0, \dots, B_{l-1}) with $l > 2$, the associated curve to be K , and suppose we are attempting to expand it in the “ B_0 to B_1 ” direction. The expansion in this direction is possible if and only if a branch of B_{l-1} lying on K and different from B_{l-2} there exist. If such a point does not exist, the expansion in this direction is no longer possible and a new attempt of expansion is performed in the “ B_1 to B_0 ” direction. The expansion from a seed set ends when the maximum expansion is reached in both the directions. Figure 3 illustrates an example of centroid network (a) and the paths identified by the algorithm (b).

3.2 Pattern Assessment

A path of centroids gives an outline of the REs placement but it is necessary to verify the correct orientation of the corresponding REs to assess that the identified path really indicates a regular pattern of repeated parts. This phase of the algorithm has been

developed for REs containing exclusively planar and cylindrical faces and for the following types of pattern: linear translational, circular translational, circular rotational and reflectional. It is based on the verification that the entities of the REs satisfy the same transformation rule of the related centroids. Specifically, for centroids lying on a linear path the candidate pattern is the linear translational, whilst for centroids lying on a circular path, the candidate patterns are the circular translational or rotational. Reflectional patterns are verified only in case of GSs grouping only two REs.

Therefore, for any pair of REs corresponding to two consecutive centroids in the path, two levels of check are performed. The first check considers the real vertices in the REs, the second exploits the surface information of the faces. The set of vertices of the REs checked during this phase does not correspond to the set of vertices used for the centroid computation. Here we exclude the two points added on closed curves since they are not granted to be inserted in the same position in different REs. Indeed, to determine these points we used the lying curve parametrization and different ways of designing two repeated entities could lead to different parametrizations.

Concerning the check of the vertices, the method verifies if for any vertex in one RE, there is a vertex in the successive RE that is obtained applying to the first vertex the transformation under verification. For example in case of linear translational pattern in which the candidate translational vector is $w = (x_w, y_w, z_w)$, we check if for every $V_{i,j}$ vertex of the repeated entity A_i , with $j = 0, \dots, p-1$, p being the number of vertices, with vertex coordinates $(x_{V_{i,j}}, y_{V_{i,j}}, z_{V_{i,j}})$, there exist $k \in \{0, \dots, p-1\}$ such that $V_{i+1,k}$, vertex of A_{i+1} has coordinates $(x_{V_{i,j}} + x_w, y_{V_{i,j}} + y_w, z_{V_{i,j}} + z_w)$.

If the check on the vertices is positive, then a second level of verification is performed on the face orientation. In case of linear translational pattern, for each planar face F_k in the first RE we verify if there exist a planar face F_h in the successive RE such that $n_k = n_h$ where n_k, n_h are the normal vectors of the planar faces. For each cylindrical face we check if there exists a corresponding translated cylindrical face in the second RE by exploiting axis and edge information. If it is verified that for $i = 0, \dots, q-1$ the REs A_i and A_{i+1} are related by translational function with the translational vector w , the sequence of (A_0, \dots, A_{q-1}) constitutes a linear translational pattern of length q . Analogously, the verification is performed for circular translational, circular rotational and reflectional pattern.

3.3 Pattern Detection in Assembly Models

In case the REs correspond to instances of the same part in a CAD assembly model, the previously described method is applied by considering as characteristic references for the RE the associated reference frame origin and axes. Therefore, the computation of the candidate patterns (i.e. the detection of possible paths of points) is applied to the points representing the origins of the reference frames, grouped in d -adjacency matrices, while the assessment of the correct orientation of the corresponding REs is performed by verifying the correspondence of the axes of the frame.

4 Implementation and Experiments

The application was developed as a plug_in of the CAD system SolidWorks [8] by exploiting its application programming environment in which B-Rep models can be created, visualized, accessed and processed. The algorithm has been developed in the C# programming language, using Visual Studio as development environment.

The algorithm has been tested on several B-Rep models both created using SolidWork, and collected from the public repositories GrabCAD [9] and TraceParts [10]. Figure 4 illustrates a small subset of the tests performed. In the example of Fig. 4 (a) the algorithm detects 1 linear translational pattern of length 3 whose components are the slots A_1 , A_2 and A_3 . The REs A_0 and A_4 are correctly excluded: A_4 because its centroid is not part of the identified pattern, while A_0 is excluded after the verification of the faces' orientation, even if its centroid is on the linear translational path. For the part in Fig. 4(b) the algorithm detects 4 linear translational patterns of length 3; the first two patterns are detected on the top side of the base component: the algorithm initially identified a path of 7 centroids but after the check on the faces the RE A_0 is excluded and the algorithm detects two distinct translational patterns. The other 2 linear translational patterns are identified on the front and right side of the base component. For the cylindrical mechanical component represented in Fig. 4(c) each RE is composed by a

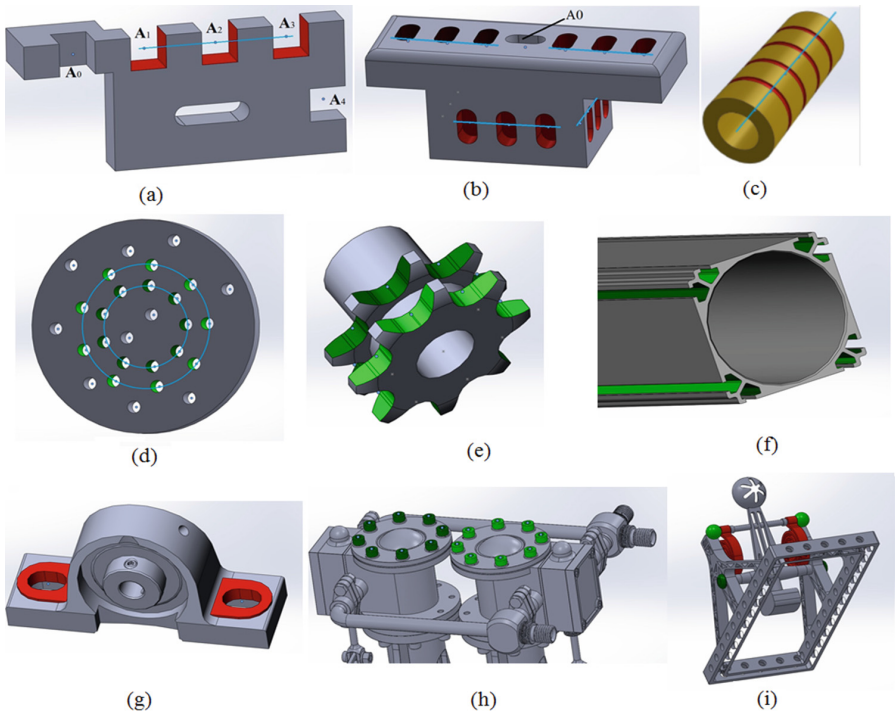


Fig. 4. A subset of the test results (Color figure online).

cylindrical face and two planar faces adjacent to it. As expected, the algorithm detects a single translational linear pattern constituted by all the 4 REs. In this case the maximum pattern has been identified.

In the example in Fig. 4(d), 28 through-holes constitute the input set of REs. In this case, the algorithm detects 2 concentric circular translational patterns of length 9 as indicated; in this case the detected patterns are not visually evident.

In Fig. 4(e) among the 16 REs composed by 3 planar and 4 cylindrical faces, 2 circular rotational patterns of 8 elements are identified. In Fig. 4(f) there are 8 REs each one made up by 6 planar and 5 cylindrical faces: 2 rotational patterns of 4 REs with the same center and radius are recognized.

In the last row of Fig. 4, in (g) an example of identification of a reflectional pattern composed by two REs is depicted; in (h) and (i) the algorithm is applied to two different assembly models and discovers respectively two rotational patterns of 8 elements in the first case, and 3 patterns of 2 elements in the second case, more specifically 2 rotational (in green in Fig. 4(i)) and 1 translational patterns (in red in Fig. 4(i)).

5 Conclusions

This paper proposes an approach to detect regular patterns of congruent sub-parts in B-Rep part and assembly models, aiming at recovering the high-level information embodied in intentionally incorporated symmetries in a given B-Rep model. The approach has been implemented in a commercial CAD system and detects linear, circular translational, rotational and reflectional patterns of congruent sub-parts of the model. Analogously to the method proposed in [7], it exploits specific characteristic points to identify regularities in repeated sub-parts, but it results more flexible as it does not need any a priori knowledge concerning the correspondence between the B-rep vertices of the repeated entities and it is applicable also to assembly models.

The proposed approach gives precedence to the detection of patterns involving the highest number of elements and it has been chosen to identify first patterns with lower distances between elements, considering the proximity a valid indicator for the pattern existence. Anyhow, in some cases it may be preferable to consider set of patterns covering the entire set of repeated entities, instead of the longest ones but not including all the sub-parts. Thus, a possible extension of the method could be its parametrization, allowing the user to indicate which criteria have to be privileged. Another option could be to extract all the possible alternative patterns and letting the user choose among them. Another possible improvement is the possibility of removing the limit of the pattern search only within a grouping surface, including also symmetric arrangements of congruent sub-parts lying on faces with different host surfaces. Furthermore, the method needs to be extended also to include conical, spherical, toroidal and freeform surfaces as well. It would be useful to extend the class of symmetric arrangements to detect, including for example also glide reflection and screw. Finally, ongoing work addresses the identification of the correlation between the detected patterns, such as for instance two circular translational patterns sharing the same circumference center and lying on the same plane or patterns of patterns.

References

1. Pauly, M., Mitra, N.J., Wallner, J., Pottmann, H., Guibas, L.J.: Discovering structural regularity in 3D geometry. *ACM Trans. Graph.* **27**(3), art. no. 43 (2008). doi:[10.1145/1360612.1360642](https://doi.org/10.1145/1360612.1360642)
2. Mitra, N.J., Pauly, M., Wand, M., Ceylan, D.: Symmetry in 3D geometry: extraction and applications. *Comput. Graph. Forum* **32**(6), 1–23 (2013)
3. Ullmann, D.G.: *The Mechanical Design Process*, 2nd edn. McGraw-Hill, New York (1997)
4. Li, K., Foucault, G., Léon, J.-C., Trlin, M.: Fast global and partial reflective symmetry analyses using boundary surfaces of mechanical components. *Comput. Aided Des.* **53**, 70–89 (2014)
5. Jiang, J., Chen, Z., He, K.: A feature-based method of rapidly detecting symmetries in CAD models. *Comput. Aided Des.* **45**(8–9), 1081–1094 (2013)
6. Li, M., Langbein, F.C., Martin, R.: Detecting approximate symmetries of discrete point subsets. *Comput. Aided Des.* **40**(1), 76–93 (2008)
7. Li, M., Langbein, F.C., Martin, R.: Detecting design intent in approximate CAD models using symmetry. *Comput. Aided Des.* **42**(3), 183–201 (2010)
8. <http://help.solidworks.com>. Solidworks industrial designer. Accessed 22 Apr 2015
9. <http://www.grabcad.com/>. GrabCAD Workbench. Accessed 22 Apr 2015
10. <http://www.tracepartsonline.net/>. Product content everywhere. Accessed 22 Apr 2015