A machine learning approach to reverse engineering based on clustering and approximate implicitization

Andrea Raffo, Oliver Barrowclough, Georg Muntingh and Vibeke Skytt

andrea.raffo@sintef.no

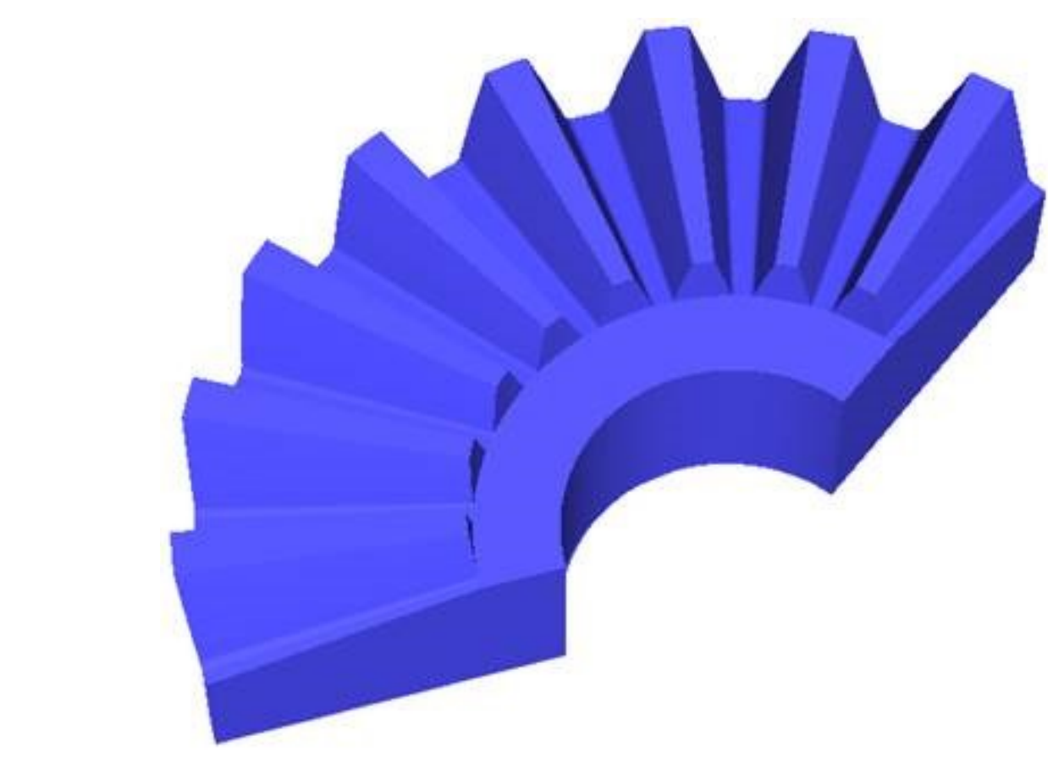
SINTEF DIGITAL, Department of Mathematics and Cybernetics, P.O. Box 124 Blindern, NO-0314 Oslo, Norway

Marie Skłodowska-Curie Actions

Introduction

In industrial applications like computer aided design, geometric models are often represented numerically as polynomial splines or NURBS, even when they originate from primitive geometry. For purposes such as redesign, it is possible to gain information about the underlying geometry through reverse engineering.

In this work we combine **clustering methods** with **approximate implicitization** to determine these primitive shapes and extract their features.



An illustrative example, shown to the left, is a simplified part of a NUGEAR developed by STAM S.r.l., Genova. This gear is used in the CAxMan project [1] and it is made up of individual patches belonging to an overarching cylindrical/conical structure.

Clustering

Clustering is an **unsupervised machine learning** technique, gathering a group of objects into a certain number of classes (or clusters). The grouping is performed so that objects in the same class are more similar to each other than to elements of other classes.

For most algorithms one needs to define a notion of distance, or more generally **dissimilarity**, between classes. A dissimilarity measure d is a positive semidefinite symmetric map assigning a real number to any pair of objects (or clusters of objects): it is a map such that $d(i,j) \geq 0$ and $d(i,j) = d(j,i)$ for all clusters i, j.

We consider two types of clustering algorithms:

- **Hierarchical clustering** [4]. In the agglomerative approach, a sequence of irreversible steps is taken to construct a hierarchy of clusters. A convenient formulation, in dissimilarity terms, is the *Lance Williams dissimilarity update formula*. Start with a cluster for every object. If the clusters i and j are joined into cluster $i \cup j$, then the new dissimilarity between the new cluster $i \cup j$ and all other clusters k is:

  $$d(i \cup j, k) = \alpha_i d(i,k) + \alpha_j d(j,k) + \beta d(i,j) + \gamma |d(i,k) - d(j,k)|$$

  where $\alpha_i, \alpha_j, \beta$ and γ specify the agglomerative criterion.

  Advantages.
  - Agglomerative hierarchical clustering algorithms can be characterized as greedy, in the algorithmic sense, making them fast.
  - The setup allows for many variations of the coefficients and similarity measure (e.g. single linkage, complete linkage, etc.).

  Disadvantages.
  - It can be difficult to specify a good stopping criterion.
- **Centroid-based clustering**. Given k cluster centers, at each step each object is assigned to the nearest cluster center.

  Advantages.
  - Balanced cluster size, both with respect to variance and number of elements.

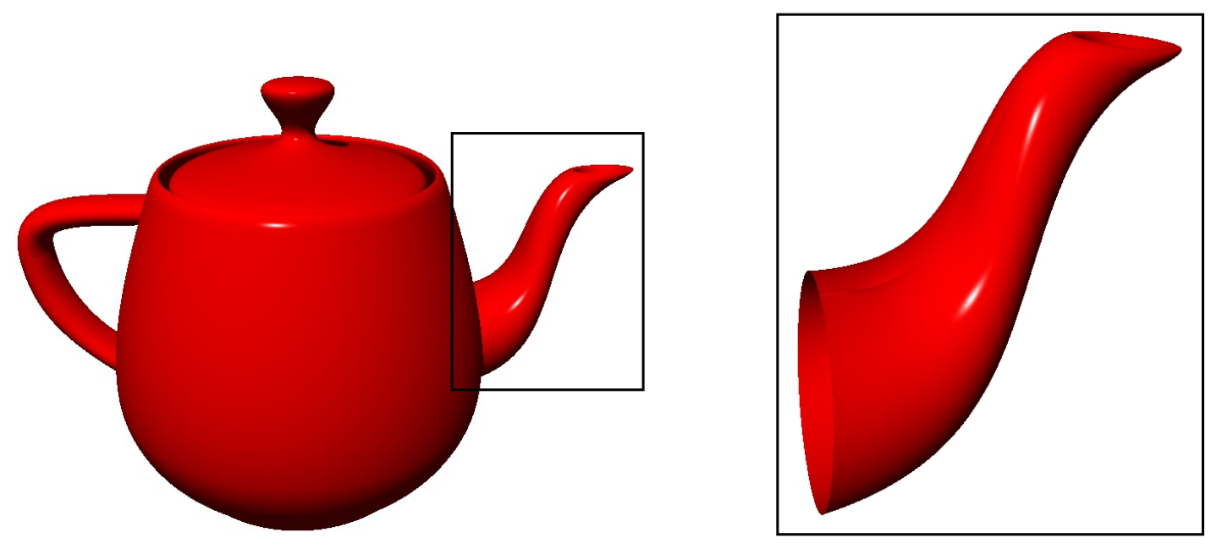
  Disadvantages.
  - One needs to specify the number of clusters in advance.
  - Results depend on the initial choice of the cluster centers.

Exact and approximate implicitisation

It is well known that for a parametric rational hypersurface (here: curves in 2D or surfaces in 3D) it is possible to compute the implicit form in a process called **implicitisation**. Since the two representations are complementary, several methods for the passage between them have been developed through the years.

- In **elimination theory**, the problem of implicitisation is solved by the elimination of the parametric variables. The result is a curve or surface represented by a single polynomial. There are different computational challenges, such as the presence of **additional solutions** and a low **numerical stability**.
- In **approximate implicitisation** (see [2] and [3]), new algorithms for ''accurate'' single polynomial approximations are introduced in common CAGD tools.

Approximate implicitisation can be performed piecewise by dividing the model into smooth components. This approach is of interest in applications such as computer graphics, where the models are rarely described by a single polynomial.



- **Discrete approximate implicitisation based on the Lagrange basis** [3].
  - Fix a basis $\Pi = [\pi_1, \ldots, \pi_m]$ for the space of implicit polynomials and a finite sequence of points $P = [p_1, \ldots, p_n]$, typically sampled from a parametrization $\mathbf{p}$.
  - Form the collocation matrix $M = [\pi_j(p_i)]_{i=1,j=1}^{n,m}$.
  - Compute the singular value decomposition $M = USV^T$.
  - Pick out the right singular vector $v_{\min} = \mathbf{c} = [c_1, \ldots, c_m]$ corresponding to the smallest singular value $\sigma_{\min}$, and form the implicit form $q = c_1\pi_1 + \cdots + c_n\pi_n$.

  This choice yields a bound

  $$\min_{\|\mathbf{c}\|_2=1} \max_{t\in\Omega} |q(\mathbf{p}(t))| \leq \Lambda \cdot \sigma_{\min}$$

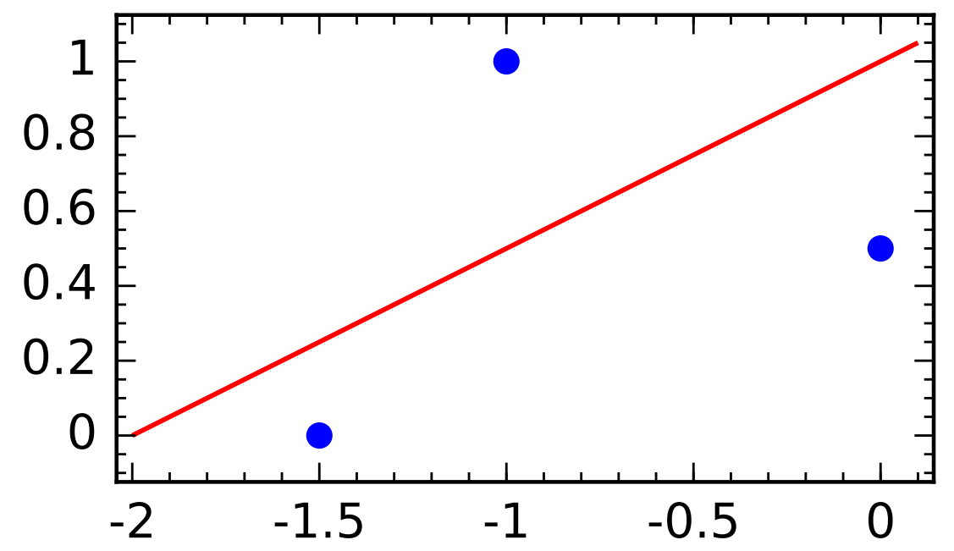
  for the algebraic error, with Λ the Lesbesque constant of the Lagrange basis. Hence $\sigma_{\min}$ is a notion of dissimilarity for the points P with respect to the basis Π.

An example of discrete approximate implicitization

For $\Pi = \{1, x, y\}$ and $P = \{(-1,1), (-3/2,0), (0,1/2)\}$ we obtain

$$\begin{bmatrix} 1 & -1 & 1 \\ 1 & -3/2 & 0 \\ 1 & 0 & 1/2 \end{bmatrix} = M = USV^T = \frac{1}{3}\begin{bmatrix} 2 & 1 & 2 \\ 2 & -2 & -1 \\ 1 & 2 & -2 \end{bmatrix} \begin{bmatrix} 5/2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/2 \end{bmatrix} \frac{1}{3}\begin{bmatrix} 2 & 1 & -2 \\ -2 & 2 & -1 \\ 1 & 2 & 2 \end{bmatrix}^T$$

and the right singular vector $1/3\,[-2,-1,2]^T$ corresponds to the implicit form $-2 \cdot 1 - 1 \cdot x + 2 \cdot y = 0$.



References

[1] Vibeke Skytt et al. *CAxMan deliverable D2.2 – IGA based shape representation, initial version*, available at `https://www.caxman.eu/mma/-/2016101914335200/`

[2] Tor Dokken, *Aspects of intersection algorithms and approximation*, PhD thesis, University of Oslo, 1997.

[3] Oliver Barrowclough, *Approximate methods for change of representation and their applications in CAGD*, PhD thesis, University of Oslo, 2012.

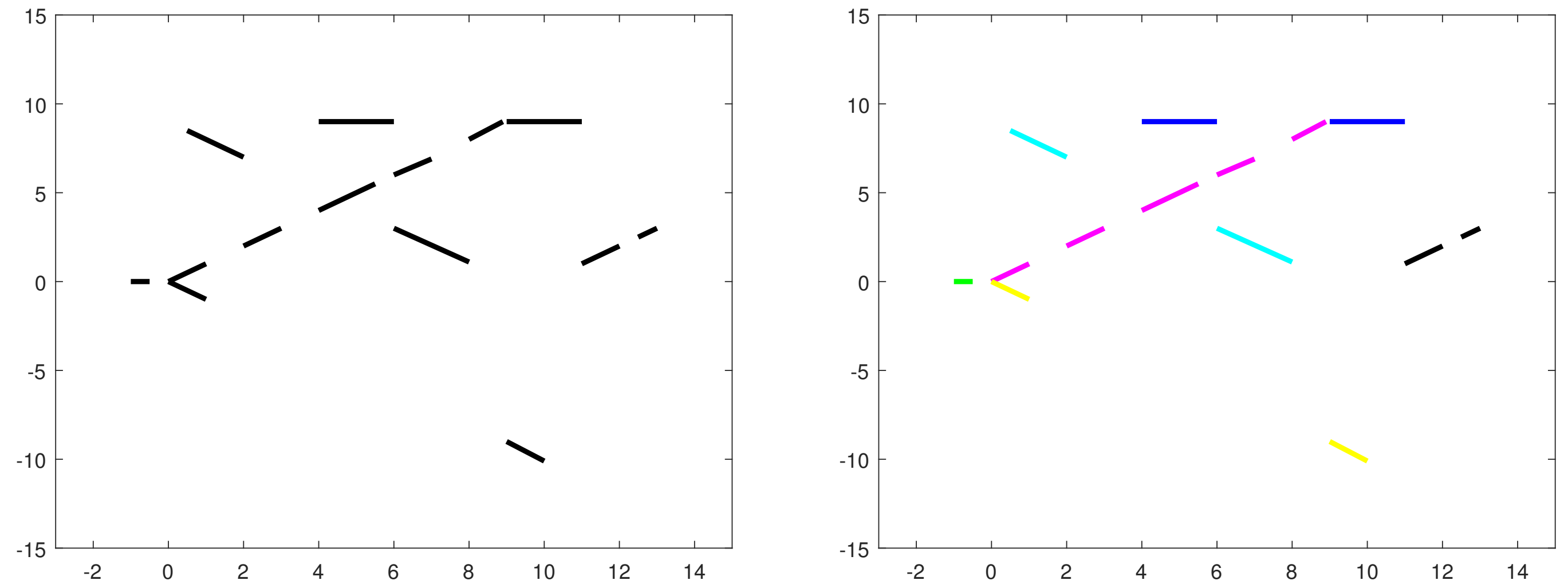
[4] Fionn Murtagh and Pedro Contreras, *Algorithms for hierarchical clustering: An overview*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. **2** (2012), 86–97.

A first toy example

As a first example, consider a set of line segments in the Cartesian plane. We wish to classify segments, clustering those that are approximately collinear.

Key steps.

- Computation of the implicit line $ax + by + c = 0$ passing through each segment (approximating when data are not exact).
- Dissimilarity of segments (or clusters thereof) assessed by the smallest singular value of the collocation matrix M.

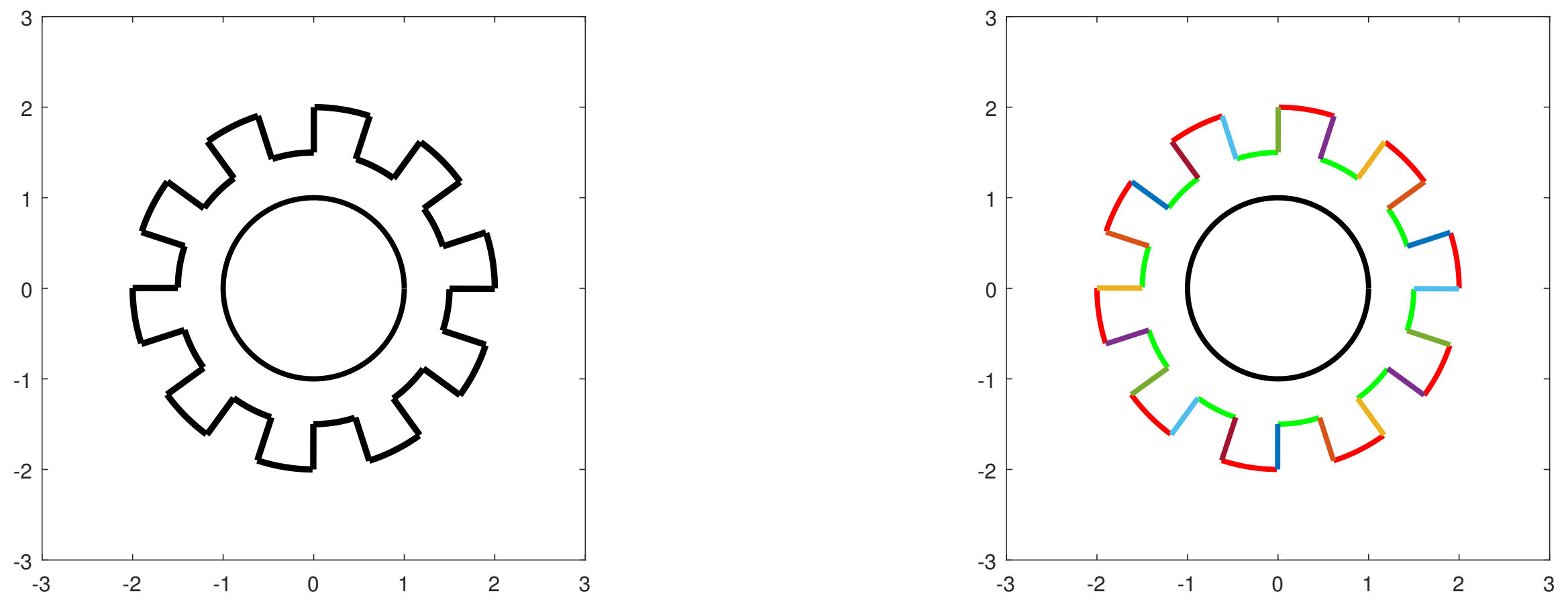


A second example: a 2D gear

A more interesting example consists of a 2D gear. We are interested in identifying curve segments approximately part of the same circle, grouping them in a cluster. The previous algorithm for line segment classification can be adapted to approximate implicitization by circles.

Key steps.

- Computation of the implicit circle $x^2 + y^2 + ax + by + c = 0$ passing through each curve segment (approximating when data are not exact).
- Dissimilarity of circle arcs (or clusters thereof) assessed by the smallest singular value of the collocation matrix M.



Conclusions and future work

The provided examples introduce a novel approach to extracting features and information on the underlying geometry of a given CAD model, by integrating approximate implicitization with clustering methods.

Future work.

- **Integration of different clustering algorithms**. Integrated use of a wider range of clustering methods (both hierarchical and centroid-based).
- **Lie sphere geometry**. Processing points, hyperplanes and hyperspheres in the same setting, using the Minkowski metric.
- **Application to natural quadrics**. Detection of cylinders and cones.