International Journal of Shape Modeling

APPROXIMATE GEOMETRIC REGULARITIES

F. C. LANGBEIN
Department of Computer Science, Cardiff University,
PO Box 916, Cardiff, CF24 3XF, UK,
F.C.Langbein@cs.cf.ac.uk.

B. I. MILLS
Institute of Information and Mathematical Sciences, Massey University at Albany,
Private Bag 102 904 North Shore Mail Centre, New Zealand,
B.I.Mills@massey.ac.nz.

A. D. MARSHALL, R. R. MARTIN
Department of Computer Science, Cardiff University,
PO Box 916, Cardiff, CF24 3XF, UK,
{A.D.Marshall,R.R.Martin}@cs.cf.ac.uk.

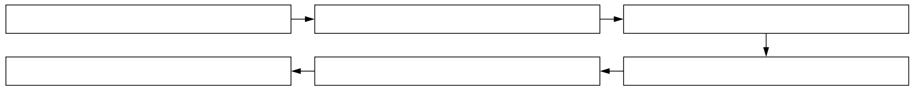
International Journal of Shape Modeling,
7(2): 129–162, 2001.

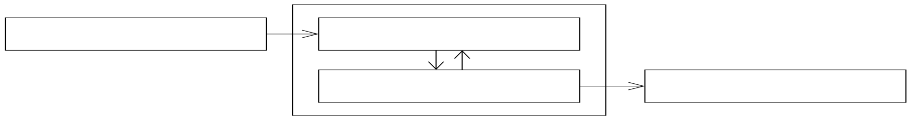
Current reverse engineering systems are able to generate simple valid boundary representation (B-rep) models from 3D range data. Such models suffer from various inaccuracies caused by noise in the input data and algorithms. Reverse engineered geometric models may be *beautified* by finding approximate geometric regularities in such a model, and imposing a suitable subset of them on the model by using constraints. Methods to detect suitable regularities for the beautification of B-rep models having only planar, spherical, cylindrical, conical and toroidal faces are presented in this paper. The regularities are described in terms of similarities. Different properties of faces, edges and vertices, and small groups of these elements in a B-rep model are represented as *feature objects*. Similar feature objects, such as directions which are parallel, form one sort of regularities. For each group of similar feature objects, *special* feature objects which might represent the group form further regularities, e.g. an integer value which approximates the radius of similar cylinders. Further regularities arise from symmetries of feature object sets. Experiments show that the regularities found are suitable for beautification such that subsequent steps allow the selection of a consistent regularity set.

Keywords: Beautification; Geometric Regularities; Geometric Constraints; Reverse Engineering; Similarity; Solid Modelling.

1. Introduction

Reverse engineering the shape of physical objects has a variety of applications in design and manufacturing, like reproduction and redesign. For many of these applications more than a simple copy is required and the information extracted from the object should represent the design intent. We are interested in reverse engi-





Element	Geometry	Feature Object	Type
face	plane	root point	position
		normal	direction
		polygonal loop(s)	loop
		root points of loops	position
	sphere	centre	position
		radius	length
	cylinder	point on axis	position
		axis direction	direction
		radius	length
	cone	apex	position
		axis direction	direction
		semi-angle	angle
	torus	centre	position
		axis direction	direction
		minor radius	length
		major radius	length
		sum of radii (unless lemon)	length
		difference of radii (unless apple)	length
edge (optional)	straight	direction of edge	direction
		distance between end points	length
	circle	radius	length
		angle of the circle segment	angle
		normal of circle plane	direction
	ellipse	normal of ellipse plane	direction
vertex	point	location	position

Table 1. Basic Feature Objects Derived from a B-rep Model

provide results of applying the methods to some example models.

2. Approximate Geometric Regularities

We describe certain geometric regularities that are approximately present in a B-rep model in terms of similarities. From a *global* point of view this leads to approximate symmetries as similarities between the model and isometric images of the model, which are discussed in [17]. This can be expanded to partial symmetries requiring that only a subset of the model is approximately symmetric or that the model can be extended in a well-defined way to make it symmetric.

Local regularities are based on properties of B-rep model elements like faces, edges and vertices, which are represented by typed *feature objects*. The *type* is defined by the property the object describes. A feature object is handled separately from the B-rep model but refers back to the element(s) which generated it. For instance, we have directional feature objects arising from the normal of a plane, and the axis of a cone or cylinder. The radius of a cylinder and the semi-angle of a cone form two other types of feature objects. Note that a single model element can generate several different feature objects of various types, which may not be independent of each other. Further feature objects can be derived by combining simple feature objects like the apex of a cone and the direction of its axis to form an axis feature object. Such axis feature objects may generate intersection points as further feature objects.

A list of basic feature objects is given in Table 1. The feature objects obtained from a B-rep model element depend on its geometry and its boundary. Note that we handle the feature objects arising from edges as optional since they do not always provide additional information about the model, and may sometimes create an unnecessarily large number of feature objects. We discuss the feature objects along with related regularities in detail below, and also add additional derived feature objects as appropriate.

We define and detect approximate geometric regularities in terms of similarities between feature objects. For one sort of regularity, we compare feature objects of the same type to derive sets of similar feature objects. For instance, we find parallel directions using directional feature objects. Another sort of regularity identifies special values for feature objects, by comparing them with predefined values, e.g. a length which is an integer. The feature objects are elements of a *feature space* defined by the feature object type. For instance, directional feature objects are represented as points on the unit sphere with antipodal points identified, which is a representation of the real projective plane $\mathbb{P}^2$. We seek (partial) symmetries of the feature objects in the feature space, e.g. for the directional feature objects we try to find

Directions	Parallel directions.	5
	Directions which have the same angle relative to a special direction.	4
	Symmetrical arrangements of directions.	4
Axes	Aligned axes.	3
	Axes intersecting in a point.	3
	Parallel axes arranged along lines and grids with regular distances between them.	3
	Parallel axes arranged symmetrically on cylinders.	2
Positions	Equal positions.	2
	Equal positions under projection.	3
	Regular distances between positions arranged on a line or a grid.	3
Scalar Parameters	Equal scalar parameters.	5
	Special scalar parameter values.	3
	Simple integer relations between scalar parameters.	4
Loops	Equal loops independent of scaling.	4
Surface Types	Surface is approximately a plane or a cylinder.	—

Table 2. Common Geometric Regularities with their Estimated Frequency.

common regularities for which we present analysis methods in Table 2. The number in the last column indicates how common the particular geometric regularity is with 5 being nearly always present to 1 being rare as determined manually (except for surface types, see below).

We look for parallel directions and directions making the same angle to a special direction. For instance, the directions could all be orthogonal to a special direction which means that they lie in a plane, or they could have some other angle to the special direction which makes them lie on a cone. In addition these directions could be arranged symmetrically in the plane or on the cone as indicated in Figure 3.

Some of the directions can be associated with positions, and thus produce axis feature objects. The positions are obtained from vertices, apices of cones, centres of spheres and tori, etc. We also look for aligned axes, and their common intersection points. Furthermore, parallel axes could be arranged along lines and grids with equal distances between them or they could be arranged symmetrically on a cylinder.

For positions alone we seek equal positions, and positions which are equal when projected onto a special plane or line derived from the main directions in the model. In addition positions may be arranged regularly on a line or a grid with equal distances between them.

Scalar parameters from faces and edges are either lengths or angles. For each type separately we seek similar parameter values, and look for special values including integers and simple fractions. We also try to find simple integer relations between pairs of scalar parameters of the same type, i.e. relations of the form

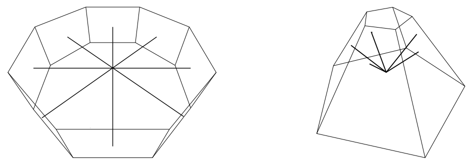


Fig. 3. Symmetrically Arranged Directions

set. We call a cluster which consists only of feature objects a *base cluster*. Each cluster ficiently distinct from each other, we do not create clusters at distinct tolerance levels. Experiments with setting a maximum tolerance lead to reasonable results, but only for simple objects was there a tolerance level which distinguished exactly between desired and unwanted regularities, and a large number of tolerances to detect different regularity types were required [15, 16]. In general the number of unwanted regularities could only be minimised, but not avoided unless desired regularities were dropped as well. As this means we have to make a decision about which regularities are used at a later stage in most cases, we drop the idea of using maximum tolerance levels and instead simplify the cluster hierarchy by determining the distinct tolerance levels within the cluster hierarchy and discard clusters above the tolerance level where the largest jump between the tolerance levels occurs. Using merit functions and geometric reasoning the subsequent steps can employ the simplified hierarchies to make consistent decisions about which regularities to include.

For simplification we use the constant Δ

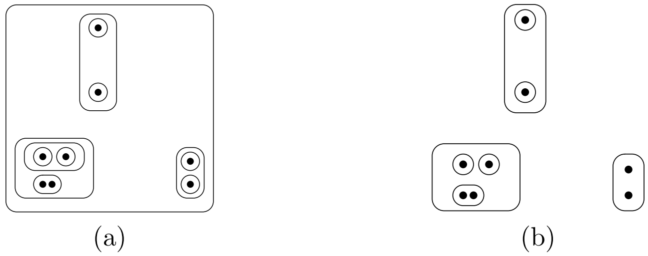


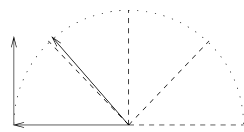
Fig. 4. Example for Hierarchical Clustering and Simplifying the Cluster Hierarchy

opposite directions, our angles are in the interval [0 senting the clusters in a least squares sense. Let hierarchy is simplified using

reported to the next higher angle cluster level until the top-level angle clusters are reached. When combining clusters at higher levels, those which are marked consistent are always added as sub-clusters to new clusters.

This results in a direction cone cluster hierarchy which is created by only comparing the directions of the cones. To avoid mixing cones with different angles, the angle clusters are used to ensure that only cone directions with similar angles are combined at different tolerance levels. The direction cone cluster hierarchy is simplified by considering the direction tolerances of the clusters using

I. Compute the angles



After the angle-regular subsets have been detected, we further check the distributions of the directions in each angle-regular subset. For a base angle the average position of the vertices around each loop, and the centre of the convex hull of each loop. Note that using several such root points leads to multiple axes for each planar face. Other possibilities exist for defining root points of planar surfaces, or for directions defined by more general curve types.

For each pair of axes we can compute an approximate intersection point as the centre of the shortest line between the two axes. Note that this point is only an approximate intersection point if the axes nearly meet. However, we consider all such points. This means an intersection point is a position we create a cluster hierarchy of approximately aligned axes. If the parallel direction cluster used to detect the parallel axes has sub-clusters, we only accept aligned axis clusters which contain axes from different sub-clusters. The resulting clusters represented as points in a plane are examined further to detect if the points lie on a line, a grid, or a circle, and are regularly spaced, as described in following sub-sections.

5.1.

Each of the point sets representing points that are approximately on a line is further examined for regular arrangements of the points on the line, i.e. we look for base distances such that the distances between a subset of the points on the line can be represented as integer multiples of a base distance, analogously to the method used for angle-regular arrangements (Algorithm 1). The main difference is that we do not have a special value such as

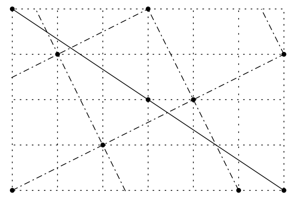


Fig. 6. Combining a Fundamental Grid with Diagonal Lines and Grids

marked with dot-dashed lines and a distance-regular diagonal line drawn solid on a fundamental grid marked with dashed lines. Only the positions of the fundamental grid which are also on the illustrated diagonals are marked.

Any distance-regular line that is not combined to give a grid or removed as a diagonal of a grid is noted as a regularity. In addition we also check whether distance-regular lines and grids have a base distance which is a special value (see Section 7).

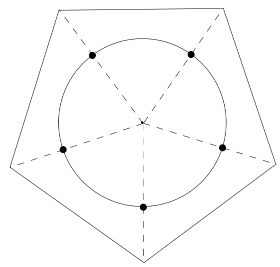
5.2.

the more possible relations between circles we take into account. We set it to Δ

which contain projected points which are as close together as the original points, as these represent approximately equal positions detected earlier. Finally the cluster hierarchy is simplified using angles we use base units

I. The function has been called as

0



computed to ensure that all relevant frequencies are considered. As the Fourier coefficients of

(a) (b) (c) (d) (e)

Time (sec): 0

gives a better picture of the actual desired regularities found and takes into account that subsequent steps have to identify regularities at high tolerance values which would require big changes to the model. Furthermore, we did not count the special values found as regularities for the results. In Figure 8 we also list the number of desired, unwanted and missed regularities found. An unwanted regularity is one that is not part of the design and conflicts with the desired regularities. Missed regularities represent major regularities not detected by our methods rather than all valid relations between the feature objects which were undetected.

Object (a) consists of two planar angle-regular arrangements of planes with base angle

By adjusting the tolerance values it was possible to include nearly all of the missed regularities mentioned above. However, this also added further unwanted regularities at higher tolerance values while the desired ones were still present at lower tolerance values. The decision about which regularities are desired has to be made by subsequent beautification steps and not by the analyser.

The experiments with simulated data showed that when using small tolerance levels we find a few, very accurate, and thus also very likely regularities. At larger tolerance levels, we detect more desired regularities and eventually all desired regularities are found. However, by increasing the tolerance levels we also increase the likelihood of detecting unwanted regularities. Only for simple objects are there tolerance levels which distinguish exactly between desired and unwanted regularities.

Because of the latter observation we do not use upper limits for the regularity tolerances, but instead we simplify the cluster hierarchy. The subsequent beautification steps have to select a suitable subset of all the regularities detected, using geometric reasoning. Often, regularities at high tolerance levels are inconsistent with regularities at lower levels. If this is detected the regularities at the higher levels can be identified as unwanted (see, for instance, the two angle-regular arrangements in object (a) which were combined to give a single angle regularity at a higher tolerance level). Besides the tolerance levels, the subsequent steps can also consider the kind of each regularity, and, for instance, prefer an orthogonal system at a higher tolerance level to setting the angles to 89°. To help make such decisions, we could also consider if a regularity would require a change in the combinatorial topological structure of the model. Additionally, we could avoid accepting regularities at extremely different tolerance levels in different parts of the object when taking a global view. However, especially if the object has been created by combining many individual views with possibly different scanner settings, the tolerance levels may not necessarily be consistent.

10.2.

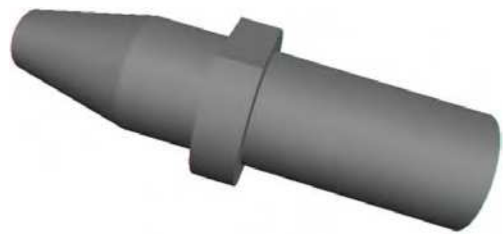


Fig. 9. A Model Reverse Engineered from Real Range Data.

clusters of another cluster. It was also found that these two planes were orthogonal to the main axis.

As the two plane directions were considered not to be parallel at a small tolerance level, some unwanted regularities were reported. There was one suggesting some special values for the angle between the directions, and a conical angle regularity consisting of the two plane directions and the main axis. Both regularities would only be realisable if the two directions are not parallel. These regularities were detected as our methods consider all sub-clusters at the different tolerance levels, in order to detect possible regularities created by them. Dependencies like this can easily be found in the subsequent steps by detecting that different values for the same angles are required by the regularities. We can either make the two planes parallel and then also orthogonal to the main axis or we accept the two unwanted regularities mentioned above. In the first case we accept a regularity at a larger tolerance level, as a parallel regularity appears to be desirable and allows other desirable regularities to be realised.

From the parallel direction cluster representing the main axes the methods further detected that the axes of the cylinders and the cone are aligned. The axis of the larger cylinder was slightly further away from the average of the aligned axes (about 0 to be conducted, though.

11. Conclusions

We have presented algorithms based on similarities to find approximate geometric regularities in inaccurately reverse engineered B-rep models. Instead of using various thresholds to decide which regularities are present in the model, the methods use cluster hierarchies, and list which regularities are present at which tolerance level. Tests with various perturbed objects were satisfactory in the sense that most desired geometric regularities were found and appear to be suitable for the subsequent beautification steps. Unwanted regularities, especially at larger tolerance levels, are also reported and will have to be identified in the subsequent beautification steps using geometric reasoning.

The methods given here could be expanded to find additional types of regularities based on the same principles. The methods could also be modified to handle other face and edge types by defining feature objects for them. Furthermore, a machining feature recognizer could be employed to partition the model into interesting subsets which could be analysed and beautified separately before they are combined using higher level beautification on the whole model. Especially for more complex models this might improve the results.

In future work we intent to develop a system which tries to find a maximal, consistent set of constraints describing the main regularities, which will be used to generate an ideal model. This includes developing methods to detect inconsistencies between the constraints based on geometric reasoning, and using optimization and graph-based methods to solve geometric constraints, while detecting inconsistencies between the constraints. We also intent to develop decision methods to resolve conflicts between contradictory regularities.

Acknowledgements

This project is supported by the UK EPSRC Grant GR/M78267. The authors would like to thank T. Várady from the Hungarian Academy of Sciences and CADMUS Consulting and Development Ltd. for providing reverse engineering software.

References