Computer-Aided Design 42 (2010) 183–201

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Detecting design intent in approximate CAD models using symmetry

Ming Li [a,b], Frank C. Langbein [a,*], Ralph R. Martin [a]

[a] School of Computer Science, Cardiff University, Cardiff, UK

[b] State Key Lab of CAD & CG, Zhejiang University, Hangzhou, PR China

ARTICLE INFO

Article history:
Received 12 March 2008
Accepted 6 October 2009

Keywords:
Design intent
Approximate symmetry
Feature recognition
Beautification
Reverse engineering
CAD data transfer

ABSTRACT

Finding design intent embodied as high-level geometric relations between a CAD model's sub-parts facilitates various tasks such as model editing and analysis. This is especially important for boundary-representation models arising from, e.g., reverse engineering or CAD data transfer. These lack explicit information about design intent, and often the intended geometric relations are only approximately present. A novel solution to this problem is presented based on detecting *approximate local incomplete symmetries*, in a hierarchical decomposition of the model into simpler, more symmetric sub-parts. Design intent is detected as congruencies, symmetries and symmetric arrangements of the leaf-parts in this decomposition. All elementary 3D symmetry types and common symmetric arrangements are considered. They may be present only *locally* in subsets of the leaf-parts, and may also be *incomplete*, i.e. not all elements required for a symmetry need be present. Adaptive tolerance intervals are detected automatically for matching inter-point distances, enabling efficient, robust and consistent detection of approximate symmetries. Doing so avoids finding many spurious relations, reliably resolves ambiguities between relations, and reduces inconsistencies. Experiments show that detected relations reveal significant design intent.

1. Introduction

Design intent concerning the *shape* of a CAD model can be expressed via geometric properties of, and relations between, its vertices, edges, faces and sub-parts. As shape is often essential to function, such relations must be enforced on the model to fulfil its purpose. Many intentional geometric relations form *geometric regularities*. However, information about a model's design intent is not always explicitly available. E.g., reverse engineering [1] captures the shape of a model but does not explicitly detect intended regularities. Such models are approximate due to measurement errors, and approximation and numerical errors occurring during reconstruction. Similarly, models constructed from inexact user input, e.g. sketches [2,3], are also approximate, and lack explicit design intent. Exchanging models between different CAD systems [4] may break intended, exact regularities due to incompatible tolerance systems and representations; design intent is often not explicitly transferred. Detecting design intent in such approximate models can reveal high-level information that is necessary for the model's function or purpose. Such information may be used to constrain and guide editing operations. It may also allow us to improve an approximate model by enforcing intended regularities. It may enable faster analysis and more compact representation, if the model has symmetric sub-parts. It may also allow models to be more meaningfully indexed for shape search, etc. Thus, this paper considers algorithmic detection of geometric design intent in approximate boundary-representation (B-rep) models of engineering objects, such as the one in Fig. 1.

Symmetry is a key concept in design. Engineering objects often exhibit symmetries for functional, aesthetic, and manufacturing reasons [5,6]. Many regularities can be represented via symmetries [7]. A symmetry is an isometry that maps a set exactly onto itself. However, symmetry may be present *approximately*—the set is almost invariant under an isometry, *locally*—only part of the set is invariant, *incompletely*—not all elements building a symmetry are present, and *compatibly*—multiple subsets share the same symmetry. We thus later define a precise concept of *approximate incomplete symmetry* which includes exact and global symmetries as special cases, generalising the ideas in [8]. For brevity, henceforth, we refer to *approximate* symmetry or congruency as symmetry or congruency, unless stated otherwise. An alternative approach [9] considers *asymmetries* in a model to describe design intent as a sequence of symmetry breaking operations.

Complex models often exhibit far too many alternative plausible approximate regularities for *exhaustive* methods to be able to determine which regularities represent the original design intent of the whole model [10]. As a simple example, consider a rectangular block with many prisms attached to its faces. Analysing the

* Corresponding author. Tel.: +44 29 20870110.
E-mail addresses: eMing.Li@gmail.com (M. Li), F.C.Langbein@cs.cardiff.ac.uk (F.C. Langbein), Ralph.Martin@cs.cardiff.ac.uk (R.R. Martin).

0010-4485/$ – see front matter
doi:10.1016/j.cad.2009.10.001

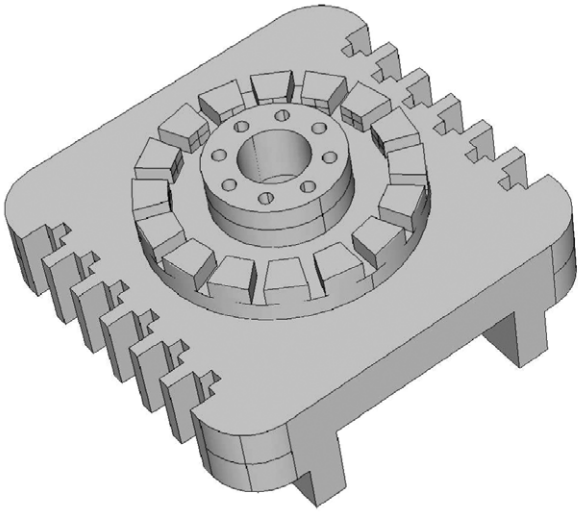

Fig. 1. An example of an approximate CAD model: Monster.

whole model without finding the prisms creates many candidate angles and distances forming plausible regularities between the model's planes. By first identifying the individual prisms as sub-parts, we can detect their approximate prismatic symmetries, and separately determine symmetric arrangements of the prisms on the block. Analysing *sub-parts* of the model separately increases the speed of regularity detection *and* provides more reliable results. Hence, our design intent detection algorithm performs model decomposition before detecting regularities in the resulting sub-parts.

The decomposition phase builds a *regularity feature tree* (RFT) forming a hierarchy of *regularity features*: simple, closed volumes which in combination describe the original shape. The regularity features at the leaves of the RFT describe the complete shape of the object; the tree indicates how to build the complete model from the leaf-parts. Unlike a CSG tree, the RFT does not contain standard primitives, nor does it give a Boolean decomposition [11]. Instead, the emphasis is on the fact that the leaf-parts are simpler and more symmetric than other parts in the tree, and not on how the object was or might have been constructed. The second phase of the algorithm seeks regularities within the model in terms of congruencies, incomplete symmetries and symmetric arrangements of these leaf-parts. It first detects congruencies to partition the leaf-parts into *congruence sets*, each containing one or more congruent leaf-parts. Next, for each congruence set, it seeks subsets forming incomplete symmetries and incomplete symmetric arrangements. Compatible symmetries shared by leaf-parts, and symmetric arrangements, are further combined before we output all detected regularities as transformations matching sub-parts of the model. The process is illustrated in Fig. 2 for the model in Fig. 1. Fig. 2(a) shows the computed RFT, Fig. 2(b) shows the congruent leaf-parts found, and the detected symmetries are given in Fig. 2(c)–(e). The output may, e.g., be used to describe a model by geometric constraints [12], or be processed by regularity selection techniques [10,13].

As the models are approximate, the method has to consider tolerances carefully. We compute suitable (tolerance) *validity intervals* directly from distances present in the model to ensure that model entities match unambiguously (i.e. in a one-to-one manner) at any tolerance in the interval. During decomposition, each different validity interval yields a different, well-defined RFT. We let the user select a suitable RFT, which is often straightforward as appropriate tolerances are often known. Regularity detection is then restricted to that particular validity interval. For a particular decomposition, regularities may also exist at different tolerance levels. To avoid missing any important regularities, we seek all of these. We ensure that the regularities are unambiguously present to avoid inconsistencies between regularities and to reduce the number of spurious regularities found. Regularities are detected in a certain sequence for efficiency, and to ensure that relations between regularities are preserved (e.g. congruent sub-parts must have the same symmetries). Tolerance information is used to ensure that these inter-regularity relations are preserved at the tolerance intervals at which the regularities are present.

Throughout this paper, we assume that the input model is a manifold 3D solid represented by a valid, watertight B-rep data

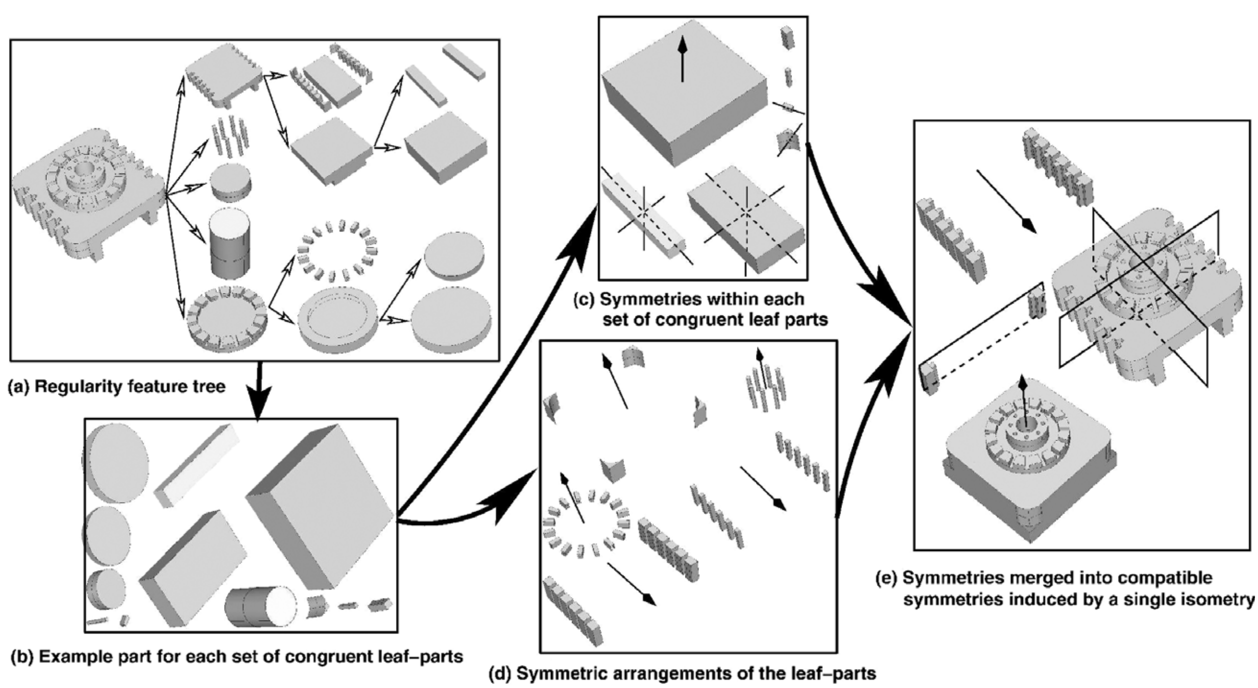

Fig. 2. Overview of algorithmic steps for detecting design intent of the Monster model in Fig. 1.