Available online at www.sciencedirect.com
ScienceDirect
Computers & Graphics 30 (2006) 903–916
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

A 3D object classifier for discriminating manufacturing processes

Cheuk Yiu Ip[1], William C. Regli*

Geometric and Intelligent Computing Laboratory, Department of Computer Science, College of Engineering, Drexel University, Philadelphia, PA 19104, USA

Abstract

Automated classification of artifacts produced by mechanical computer-aided design (CAD) is a unique research frontier for 3D matching and mesh processing. Unlike general graphical models, mechanical CAD artifacts have a physical realization via a variety of manufacturing processes as well as functional and behavioral attributes. The general problem of how to best correlate low-level shape data with the higher-order manufacturing and mechanical properties remains an open area of research with many practical applications (cost estimation, design archival, variational design and process selection).

This paper addresses the problem of manufacturing process discrimination, i.e., determination of the best (or most likely) manufacturing process from shape feature information. Specifically, we introduce a new curvature-based shape descriptor and show its applicability to manufacturing process discrimination using a publicly available set of artifacts from the National Design Repository. Statistics on surface curvatures are used to construct the curvature-based shape descriptor; and a supervised machine learning classifier, based on support vector machines, is applied to learn a separator for models that are "prismatic machined" and "cast-then-machined". The authors believe that this work can be the basis for practical new techniques for manufacturing cost estimation, engineering analysis and design archival.

Keywords: 3D search; Shape recognition; Shape matching; Solid model databases

1. Introduction

This paper examines the manufacturing classification of mesh-based Computer-Aided Design (CAD) models with curvature descriptors and support vector machines (SVMs). Previous research, such as feature recognition and process planning, focused on extracting manufacturing information from 3D solid models of CAD objects. Mesh models have become a useful CAD representation thanks to the development of rapid prototyping and 3D scanning acquisition technologies. Mesh models provide a simple, uniform representation that is easily preserved and transformable. Working with meshes allows comparisons of models generated by any CAD system. The issue with mesh-based representations is that they capture only geometry and topology and little, if any, of the manufacturing or design semantics that would be useful in answering engineering queries.

A long-term goal of our work is to develop methodologies to interact with CAD data in engineering information management systems and enable long-term preservation of engineering artifacts. This paper bridges the gap between low-level shape representation and engineering semantics by presenting a methodology for discriminating the manufacturing processes for an individual part solely from the mesh representation of the artifact. Machining of discrete parts is a fundamental manufacturing process in aerospace, automotive and other industries. The machining process consist of material removal operations (i.e., drilling, milling, etc.) on a piece of stock material. Discrete parts that are exclusively machined are usually high-precision parts or parts made in small batches (i.e., for custom jobs). For larger production runs machining is not cost effective. In these cases, part stock shapes may be created using a casting process and then the finishing features are machined. Cast-then-machined parts are

*Corresponding author. Tel.: +1 215 895 6827; fax: +1 215 895 0545.
E-mail addresses: ipcy@umd.edu (C.Y. Ip), regli@drexel.edu (W.C. Regli).
[1]Present address: Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA.

0097-8493/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2006.08.013

typically for larger production runs and generally have much looser tolerance considerations for the non-machined surfaces of the object. In this case the investment of the physical plant is larger, as is the manufacturing production plan (i.e., one needs to machine a mold with which to do casting), but the per-unit cost tends to be much less when production is in sufficient quantity. An example is given in Fig. 1(a) and (b).

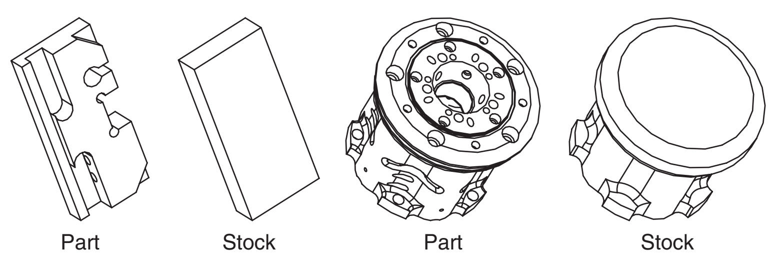
This paper introduces an adaptation of shape matching and machine learning to the discrimination of prismatic machined and cast-then-machined manufacturing categories in a database of mechanical CAD objects. To achieve this, we develop a new shape discriminator based on the surface curvature of an artifact and show how to use SVMs to learn the separation between the feature descriptions of the two manufacturing processes. Lastly, we provide an empirical validation with dataset of engineering artifacts and make this dataset available via the Internet to enable others to reproduce and improve upon these results.

Organization of this paper: This paper is organized as follows. Section 2 describes the general scientific challenges involved in adapting shape matching techniques to computer-aided design data, specifically data from mechanical CAD/CAM systems. Section 3 briefly overviews related work, both in CAD retrieval and in shape matching and computer vision. Section 4 presents the technical approach, detailing the development of the curvature-based shape descriptor and the use of the SVMs to discriminate among manufacturing classes. Section 5 gives an experimental evaluation of the technique, specifically comparing it with the baseline offered by using several existing state-of-the-art techniques as well as the k nearest neighbors approach. Finally, Section 6 presents discussion, conclusions and ideas for future work.

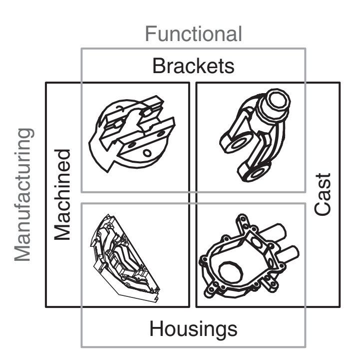
2. The challenge of CAD objects

Research on searching, classifying and comparing CAD models is an active research area, having produced a rich set of computational techniques. Existing research includes algorithms that work with photo images, projected profiles, feature interactions, and shape functions. In most cases, these techniques or systems were often presented and evaluated with their own particular datasets—datasets that contain mostly general shape models and few real CAD artifacts. This makes it very difficult to assess how effective these different techniques would be at managing CAD data. CAD artifacts and their engineering domains introduce several challenges not adequately addressed by existing research:

- *Engineering artifacts each have a physical realization.* CAD models in our dataset are manufacturable physical parts. Existing shape matching techniques, for the most part, emphasize the comparison of the gross shape of coarse artificial objects. The datasets (e.g., trees, airplanes, and boats) studied in most existing shape retrieval systems do not represent actual, or even acquired, models of physical artifacts.
- *Engineering classifications are not subjective.* In existing shape retrieval literature, datasets are pre-classified based mostly on human intuition (i.e., boats get grouped with boats; airplanes with airplanes). In contrast, engineering classifications are usually not so subjective. For example, a part is machinable on a 3-axis



(a) Prismatic-machined. (b) Cast-then-machined.



(c) Multi-classications.

Fig. 1. Machine stock material and the resulting artifacts from different manufacturing processes (a,b). An example of different engineering classifications for the same artifacts (c).

machining center or it is not; a part has four symmetrically spaced holes for fastening with bolts or it does not. This paper considers prismatic machined and cast-then-machined manufacturing artifacts, such as shown in Fig. 1(a) and (b).

- *Different valid classifications exist for the same objects.* The fact that an object may have several valid classifications is one of the fundamental problems in the field of pattern recognition. However, in engineering domains the differences across classifications can be large, and the feature set for discriminating these differences are very hard to isolate. An example is given in Fig. 1(c), which shows four parts classified two different ways (based on functional properties and manufacturing properties, respectively).

Of specific importance in this paper, we introduce a dataset of CAD artifacts classified based on the manufacturing process for creating the physical artifact. All datasets used in this paper are freely available in the National Design Repository at http://www.designrepository.org/datasets/. CAD models in the National Design Repository datasets have been collected from industry. A sample view of the National Design Repository CAD models is shown in Fig. 2. A subset of 110 parts was classified by hand into (1) prismatic machined parts and (2) parts that are first cast and then have their finishing features machined. Fig. 3 shows a sample of this dataset, and Table 1 shows a brief summary of this dataset. http://www.designrepository.org/datasets/machined.tar.bz2 and http://www.designrepository.org/datasets/cast.tar.bz2.

3. Related work

We briefly review the research work on representing and comparing 3D models. Additionally, we present a survey of some benchmark datasets from closely related disciplines like computer graphics and vision. A complete review of relevant literature in these areas is not possible due to the dynamic and evolving nature of this field. For more detailed surveys, interested readers are referred to several recent survey articles [1–3].

3.1. Representation of CAD models

Most CAD models are solid models defined parametrically. However, approximated shape models represented by a polygonal mesh are becoming another useful representation thanks to the development of rapid prototyping from approximated models and the acquisition of shape models through 3D scanning.

Solid model representations of CAD objects are traditionally exact representations of 3D solids, which are suitable for creating physical models. In commercial CAD systems like Pro/Engineer and I-DEAS, models are dominantly represented by exact parametric or different kinds of engineering features. Each objects is represented by a data structure that gives information about the object's faces, edges, vertices, and how they are joined together. For example, under a boundary representation (B-Rep), two types of information are recorded: (1) a topology record of the connectivity of faces and edges; (2) and a set of parametric equations that describes the

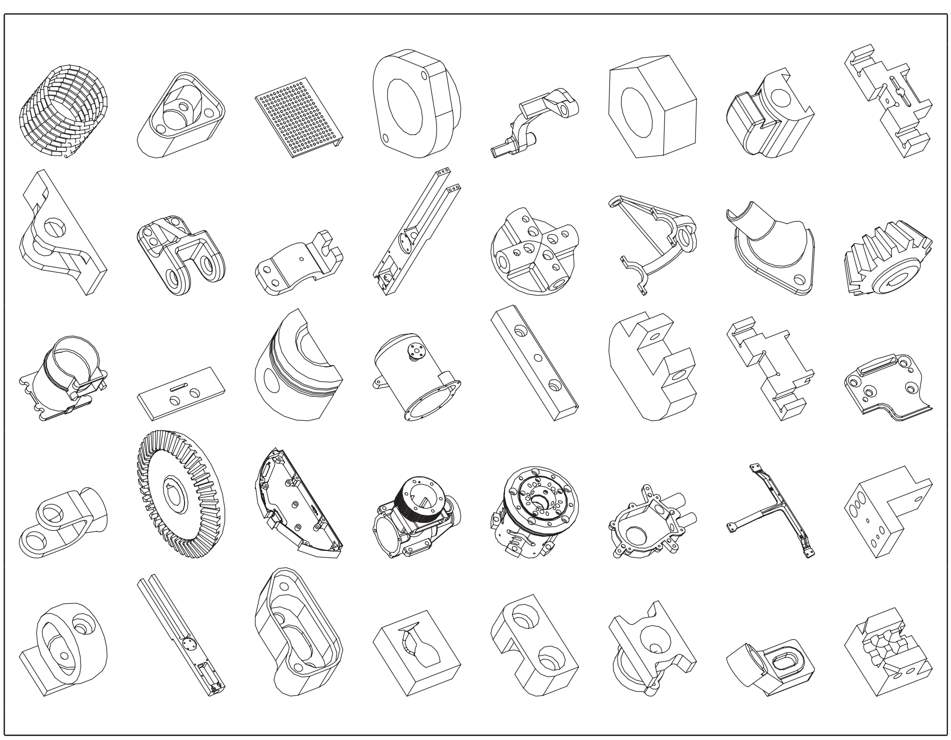


Fig. 2. Examples of 3D models from the national design repository.

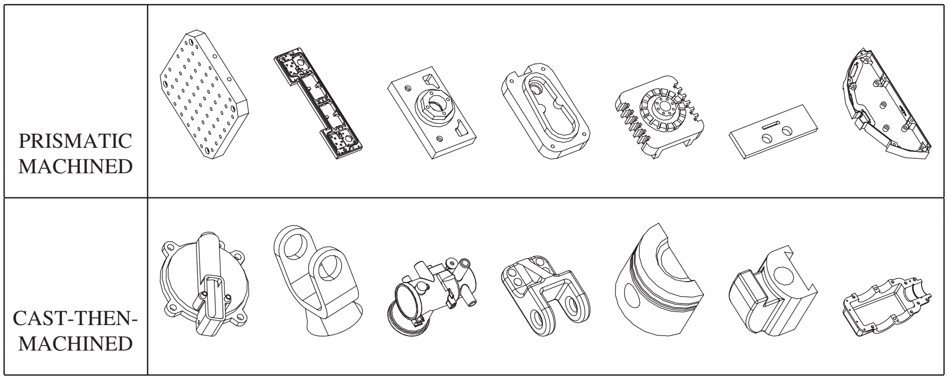


Fig. 3. Examples 3D models from the manufacturing classification dataset.

Table 1
Statistics about the 3D models in the manufacturing classification dataset

	#Models	Average #faces	Average #polygons
Prismatic machined	56	106	3600
Casted-then-machined	54	80	3447
Total	110		
	Average SAT size (KB)	Average STEP size (KB)	Average VRML size (KB)
Prismatic machined	146	233	162
Casted-then-machined	277	314	159

geometry and the location of vertices, faces, and edges (e.g., NURBS). Solid models give a complete and compact representation for design, simulation, and manufacturing purposes. Yet these models are usually stored in proprietary data formats between different CAD/CAM systems. Thus, for example, comparing models generated on I-DEAS against Pro/Engineer involves some lossy data exchange process, through conversion from STEP to IGES or approximated shape models.

Shape model representations of 3D objects are approximated models characterized by a mesh of polygons for presentation or rendering purposes in computer graphics. Rather than exact parametric equations, polygons are used to approximately curved surfaces. Only the geometry of triangles are stored without any topological information. In contrast to proprietary solid model formats, open mesh file formats such as VRML and STL, are widely available. Although shape models are not suitable for modeling physical properties or simulations in CAD/CAM systems, polygonal meshes can serve as the lowest common denominator in comparing CAD models, by faceting solid models generated by different modeling systems. Shape models of objects can also be acquired easily by using laser scanners or CT to enable comparison of digital and physical artifacts.

3.2. Comparing solid models

There are two basic types of approaches for matching and retrieval of solid based 3D CAD data: (1) *feature-based* techniques and (2) *shape-based* techniques. Some of the past work in this area is reviewed in addition to the work from computer graphics and computer vision that is related to this paper.

In practice, indexing of parts and part families had been done with group technology (GT) coding [4]. GT was designed to facilitate process planning and cell-based manufacturing by imposing a classification scheme on individual machined parts. These techniques were developed prior to the advent of inexpensive computer technology, hence they are not rigorously defined and are intended for human, not machine, interpretation. Some of the early work on feature identification from solid models aimed to find patterns in model databases or automate the GT coding process. The common aspect of all of these techniques is that they are all post priori: one runs their algorithm on model, and it produces the category or label for it.

Feature-based techniques [5–7] dating from late 1970s [8], extract engineering features (e.g., machining features, form features, etc.) from a solid model of a mechanical part for use in database storage, automated GT coding, etc. Elinson et al. [9] used feature-based reasoning for retrieval of solid models for use in variant process planning. Cicirello and Regli [10] examined how to develop graph-based data structures and create heuristic similarity measures among artifacts; this work was extended in [11] to a manufacturing feature-based similarity measurement. McWherter et al. [12] have integrated these ideas with database techniques to enable indexing and clustering of

CAD models based on shape and engineering properties. Cardone et al. [13] compared machining features of solid models for manufacturing cost estimation using solid models.

3.3. Comparing 3D shape models

The shape-based techniques are more recent, owing to research contributions from computational geometry, computer vision, and computer graphics. From the polygon mesh, different transformation invariant attributes can be extracted as the means of similarity among 3D models. Thompson et al. [14] examined the reverse engineering of designs by generating surface and machining feature information off of range data collected from machined parts. Hilaga et al. [15] present a method for matching 3D topological models using multi-resolution reeb graphs. The method of Osada et al. [16] creates an abstraction of the 3D model as a probability distribution of samples from a shape function acting on the model. Novotni and Klein [17] demonstrated the use of 3D Zernike descriptors. Kazhdan et al. [18] compares 3D models with spherical harmonics. While these techniques target general 3D models, Ip et al. [19,20] and Bespalov et al. [21] are focused on comparing shape models of CAD with shape distributions and scale-space representations. Iyer et al. [22–24] presented a CAD oriented search system, based on shape, voxelization and other approaches. Pal et al. [25] extracted features from CAD models using genetic algorithm.

3.4. Benchmark datasets

There are many benchmark datasets comprised of synthetic and realistic data in the domain of computer vision and computer graphics. The Columbia Object Image Library (COIL-100) [26] aimed to assist object recognition from 2D photos. It contains 7200 photos of 100 objects in different poses. In face recognition research, the Yale face database provides 5760 images from 10 people each seen under 576 viewing conditions for testing. A number of synthetic image sequences are provided to test optical flow and motion analysis applications. Recently, the Princeton Shape Benchmark [27] has provided 1,814 3D polygonal models, collected from the web, for evaluating shape-based retrieval and analysis algorithms. The models were chosen from heterogeneous categories ranging from animals, furniture, and airplanes.

For CAD data, the largest publicly available dataset is the National Design Repository [28,29]. From this dataset, several sub-sets have been offered to the community to test how 3D search techniques can discriminate functional classes, manufacturing objects, and human-generated classifications [30]. Objects in these datasets include mechanical CAD objects, Lego models, and objects contributed by industry and CAD vendors. Another CAD dataset is provided by Purdue University [31].

3.5. Relationship of this work to prior art

The approach and results presented in this paper contribute to advancing the field in several key areas. First, current research on shape matching techniques generally focuses on the gross shapes of mesh models. These types of techniques do not adequately discriminate among artifacts at a detailed level, such as would be required to recognize those fabricated by different manufacturing processes. This work introduces a new shape descriptor based on curvature that has a demonstrated utility in answering practical engineering queries. Second, the approach in this paper provides a detailed example of how to integrate supervised machine learning with work in shape recognition and matching. With supervised machine learning framework, one can tune a shape metric to discriminate specific classification schemes according to examples.

Recently, research from industry and academia examine the use machine learning techniques to train a 3D shape recognition system with CAD data. Work in industry [32] has explored the use of neural networks to identify parts (fasteners) based on multiple 2D views. Hou et al. [33] attempted to use shape information to cluster the semantics of parts with SVMs. In the context of shape model matching, Elad [34] used linear SVMs to adjust retrieval results from a 3D shape database according users' feedback.

Lastly, this work makes available a large, pre-classified set of 3D engineering objects. The properties of these objects are distinctly different from those in other shape retrieval datasets available over the Internet. By making these CAD objects available, the work presented in this paper is completely reproducible, and the authors hope to enable others to explore the problems specific to matching engineering objects.

4. Technical approach

Our technical approach has two major elements. First, we introduce an adaptation of existing work on surface curvature estimation to create a curvature-based shape feature descriptor. Second, using this curvature-based feature, we demonstrate how to train a SVMs classifier to discriminate across a set of 3D CAD objects belonging to one of two classes determined by manufacturing process: prismatic machined parts or cast-then-machined parts.

This research proposes the use of surface curvature and support vector machines to classify between prismatic-machined and cast-then-machined models. Surface curvature is introduced as a relevant feature for distinguishing the two processes.

Considering the limited accessibility of cutting tools in 2.5D machining processes, material removal machining operations can only construct a finite set of surfaces. In contrast, the casting process allows a larger variety of surfaces. Fig. 1 shows parts manufactured by prismatic-machined and cast-then-machined processes and their respective stocks before the part getting machined.

Without the casting process, prismatic-machined parts are manufactured from uniform stocks like rectangular and cylindrical blocks. Cast-then-machined processes cast out the stock before being machined. The stocks for cast-then-machined parts are more customized. A larger variety of surfaces are generated before machining.

The surfaces of the resulting parts hint how the parts were manufactured. Therefore, a shape descriptor focuses on local surface differences is required to classify the two process effectively. The challenges for classifying these two processes are that both processes share a set of surfaces generated by machining processes. Fig. 4 shows parts manufactured by both prismatic machined and cast-then-machined processes with their machining features (holes) highlighted.

4.1. Extracting surface curvature features

The curvature of a point on a planar curve is defined as the reciprocal of the radius of the osculating circle at that point. Extending this to regular surfaces, normal curvature, κ_n, is the curvature of the intersection on the smooth surface with a plane in direction (s, t). κ_n satisfies the following equation:

$$\kappa_n = \begin{pmatrix} s & t \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} s & t \end{pmatrix} \mathbf{II} \begin{pmatrix} s \\ t \end{pmatrix}.$$

Principal curvatures are the largest and smallest curvature at a point in all directions. The principal directions are the directions in which the principal curvatures occur. If (s, t) are expressed in the principle directions, then the formulation for κ_n becomes

$$\kappa_n = \begin{pmatrix} s' & t' \end{pmatrix} \begin{pmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{pmatrix} \begin{pmatrix} s' \\ t' \end{pmatrix} = \kappa_1 s'^2 + \kappa_2 t'^2.$$

This formulation shows κ_1 and κ_2 are the eigenvalues of second fundamental tensor $\mathbf{II}$, where (s', t') are eigenvectors. Meaning that principal curvatures and directions can be found by eigenvalue decomposition on the tensor $\mathbf{II}$.

Different representative curvature values can be computed from principal curvatures.

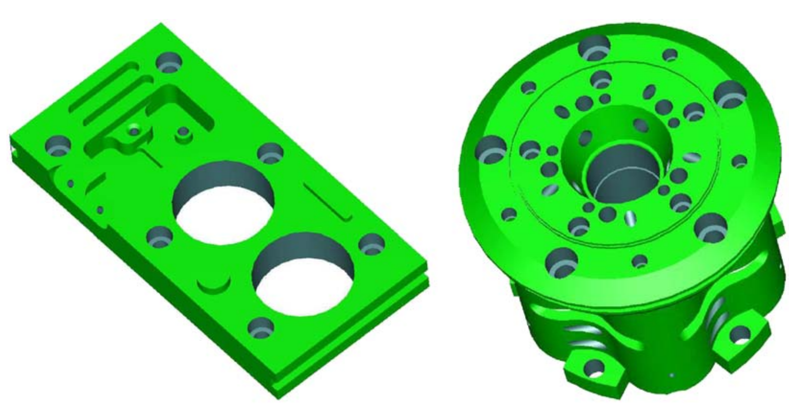
(1) Maximum curvature κ_1;
(2) Minimum curvature κ_2;
(3) Mean curvature $H = (\kappa_1 + \kappa_2)/2$; and,
(4) Gaussian curvature $K = \kappa_1 \kappa_2$.

4.1.1. Curvature estimation

Estimating curvature from mesh has been a great interest for both computer graphics and vision. Curvature information has been used in a variety of applications: mesh smoothing, repairing surfaces, crest detection, re-meshing, and non-photorealistic rendering. Exact curvature information can be computed from parametric surfaces. While mesh representation provides a piecewise approximation of surfaces, curvatures are also approximated. Some of the recent work includes: Taubin [35] and Page et al. [36] estimating the curvature tensor through a weighted average of normal curvatures of neighboring vertices. Meyer et al. [37] introduced discrete differential geometry operators. Goldfeather and Interrante [38] presented a cubic method for estimating principal directions. Lavoué et al. [39] segment meshes of CAD models by analyzing curvature tensor. This research employs Rusinkiewcz's method [40] for a fast single pass estimation of curvature from a smooth mesh.

Rusinkiewcz's algorithm works particularly well on smooth mesh models, such as those produced by faceting solid models. It estimates curvatures per vertex from its immediate neighbors. To enable curvature estimation on coarse and noisy mesh models, such as those obtained by 3D laser scanning, curvature estimation algorithms need to consider a larger geodesic neighborhood. One example is presented in [36].

Following [40], per vertex curvature is computed by weighting the curvature of triangle faces adjacent to the vertex. Assuming each triangle on a mesh surface is a



(a) Prismatic-machined. (b) Cast-then-machined.

Fig. 4. Machining is a finishing operation on many cast parts, hence artifacts manufactured by different processes may share similar features. Here, machined holes on both objects are highlighted in gray.

smooth curved surface, curvature can be computed by solving **II** with constraints setup by the normals. **II** can be expressed as the derivatives of the surface normal:

$$\mathbf{II} = \begin{pmatrix} D_u n & D_v n \end{pmatrix} = \begin{pmatrix} \dfrac{\partial n}{\partial u} u & \dfrac{\partial n}{\partial v} u \\ \dfrac{\partial n}{\partial u} v & \dfrac{\partial n}{\partial v} v \end{pmatrix}.$$

For each triangular face, three edges and normals can be used as constraints for estimating **II**. The process is then:

(1) compute per triangle face curvature by solving for the tensor **II** according to the constraints of difference between three normals;
(2) transform the coordinates with respect to the local coordinate frame of the vertex;
(3) weight the contribution of each adjacent face according the Voronoi area of the triangle; and,
(4) find the eigenvalues and eigenvectors of the tensor to determine the principal directions and curvatures.

Example. To visualize the surface differences between prismatic-machined and cast-then-machined parts, curvature values are computed for all vertices on the samples models. Fig. 5 shows sample models that are colored according to different curvature values: minimum, maximum, mean, and Gaussian. Regions with zero curvature are colored in white, non-zero curvature regions are colored according to curvature values.

4.1.2. Assessing curvature feature relevance

The following observations can be made:

- The cast-then-machined process produces artifacts with a higher portion of curved (shaded κ_1) surfaces.
- The minimum curvature, κ_2, and Gaussian curvature, K, of machining features, holes, slots, pockets, and surfaces are zero (shown in white), resulting in almost zero curvature for all prismatic machined parts.
- Cast-then-machined parts possess a higher variation of curvature values (shown using more colors).
- The difference of minimum and maximum curvatures (colors) for prismatic-machined parts is smaller than for cast-then-machined parts.

Different curvature values signify different types of surfaces on the CAD models. While both manufacturing processes produce similar surfaces through machining operations, the cast-then-machined process is expected to leave a larger variety of surfaces. Therefore, some distinctive curvature values should only be found on cast-then-machined parts but not prismatic-machined parts. These unique features separate the two manufacturing processes.

4.1.3. Defining a curvature-based shape descriptor

Statistics on per vertex curvature are used to construct the shape descriptors for classification. Curvature values naturally vary from $-\infty$ to ∞. To avoid extreme values

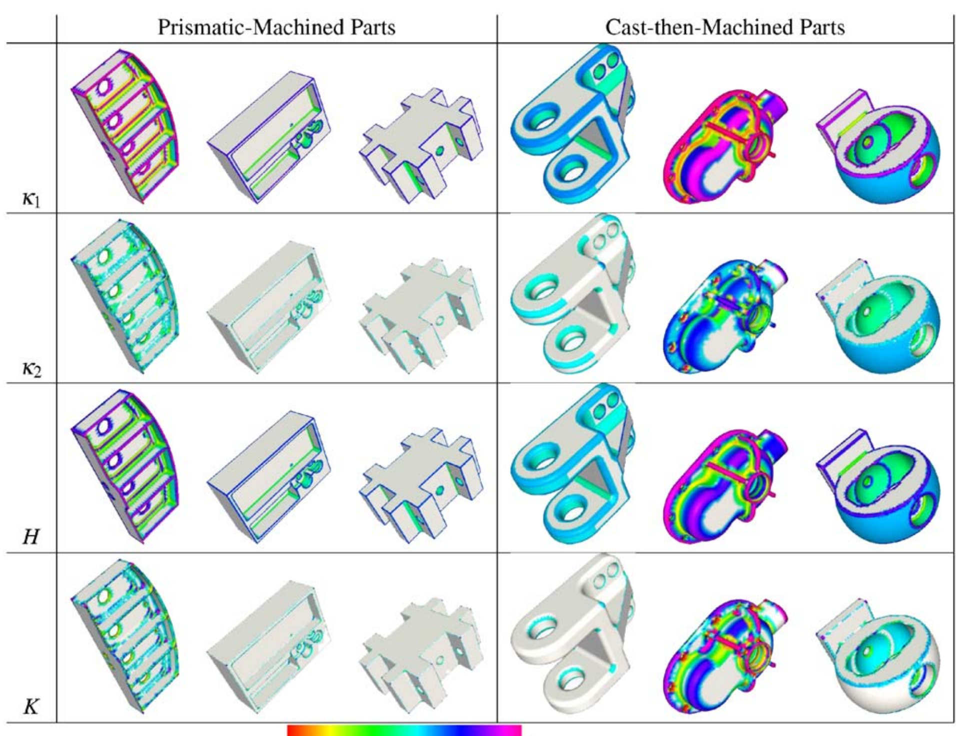


Fig. 5. 3D models color coded based on curvature values.

and numerical problems, curvature values, c, are mapped from to $[-1, 1]$, which is similar to [41].

$$c' = \frac{c}{\|c\|}\left(1 - \frac{1}{1 + \|c\|}\right).$$

Equal width bins divide the range $[-1, 1]$ to record the frequencies of different curvature c' values. Frequencies of curvature values are normalized, and curvature bins are aligned to produce meaningful comparisons. For example, frequencies of planar surfaces $[c = 0]$ always align to the same bin.

Example. Fig. 6 shows a sample of CAD models and the corresponding descriptors. As expected, cast-then-machined parts show a larger variation in terms of any curvature statistic. One might note that the center bar, $c' = 0$, often dominates the statistics in the histogram. This zero bar shows the proportion of (intrinsically, for κ_2, K) planar surfaces. The significance of this feature will be automatically determined by the machine learning classifier through the training process.

4.1.4. On the invariance of curvature statistics

Curvature statistics are a rotationally invariant feature set because the curvature values are local measures on the surface of models. Frequencies of the curvature measure are normalized to enable comparisons of models with differing numbers of vertices and different mesh resolutions. However, the curvature statistics varies with the scale of model. Since the type of surfaces are aligned in our formulation, the same model in different scales could result in different curvature statistics. Scale invariance is desirable in general context of shape matching, but it is not considered in the context of this work. For example,

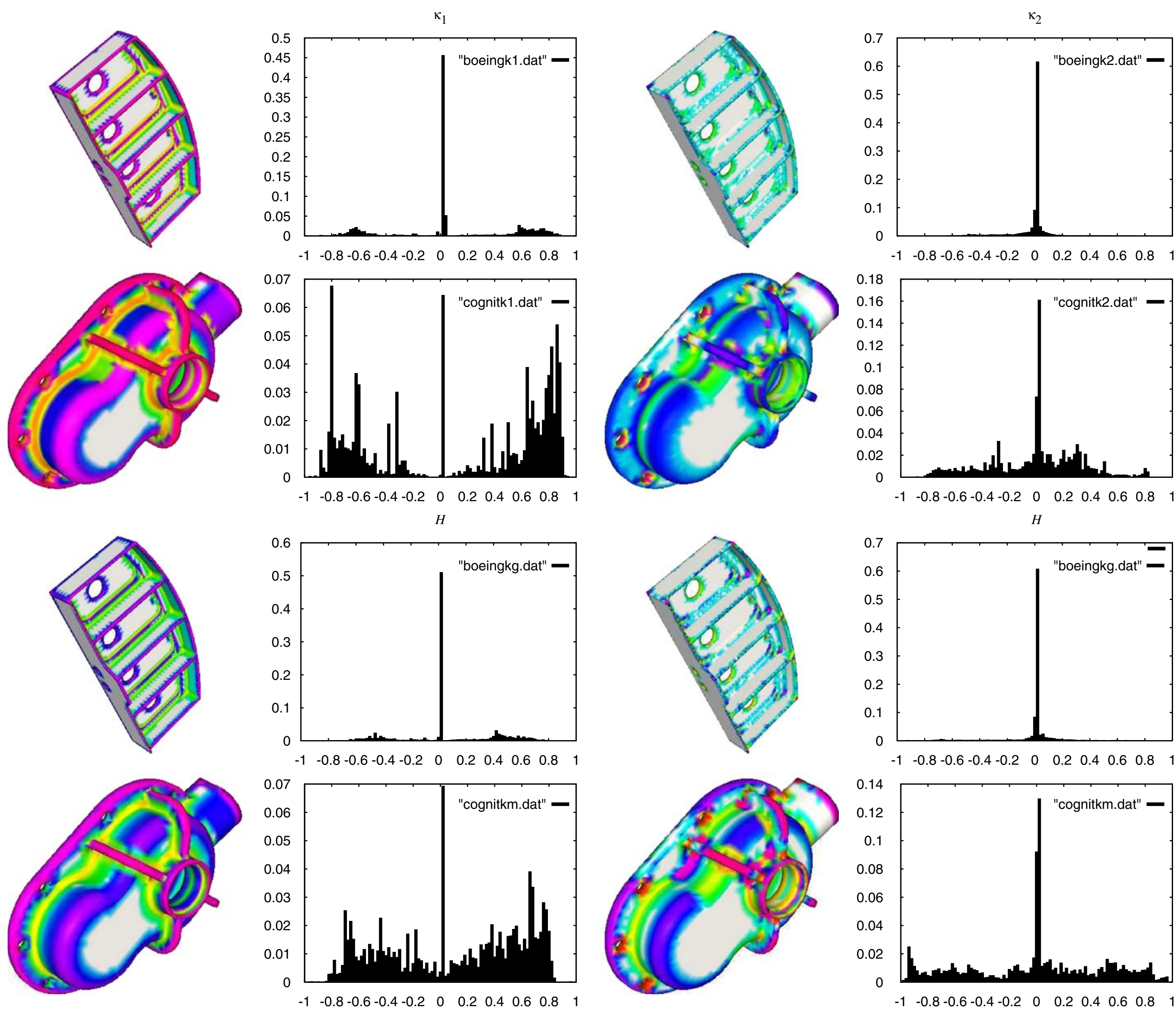


Fig. 6. Curvature shaded models and their curvature feature descriptors.

consider manufacturing two simple rings of different sizes. The smaller one consists of a hole of diameter 1 cm; whereas the larger hole diameter is 50 cm. It is likely that the smaller one is constructed by drilling, but the large one is done by a casting process. Nevertheless, if scale invariance is desirable, it can be achieved by simply rescaling the model to a fixed volume, such as the unit cube.

4.2. *Classifying objects with curvature descriptors*

Classification of prismatic-machined and cast-then-machined processes can be learned by different classifiers. SVMs with a non-linear kernel function are the choice of classifier in this research. In addition to the use of SVMs, the commonly used nearest neighbor classifier is also applied for a comparison evaluation.

4.2.1. *Primer on support vector machines*

Support vector machines are a supervised machine learning technique proposed by Vapnik [42,43]. They find the maximum margin classifier from example data in different classes. Given examples $\mathbf{x}_{1\ldots l}$, SVMs find a linear classifier that satisfies $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \geqslant 0$, with a margin of width $2/\|\mathbf{w}\|^2$. Minimizing $\|\mathbf{w}\|^2$ in the Lagrangian formulation of the classifier maximizes the width of the margin and forms the following quadratic programming problem:

$$L_P = \frac{1}{2} - \sum_{i=1}^{l} \|\mathbf{w}\|^2 - \alpha_i y_i(\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i,$$

which is equivalent to maximizing the dual of L_P

$$L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j}^{l} \alpha_i\alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j.$$

To generalize SVMs for non-linear cases, training examples can be projected into a higher dimensional space by some function $\Phi(\mathbf{x})$ for linear separation. Observe that L_D only depends on dot products in between $\mathbf{x}_i$ and $\mathbf{x}_j$, which can be substituted by a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ that computes $\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$, rather than directly computing in the high dimensional space. Assuming $K(\mathbf{x}_i, \mathbf{x}_j)$ can be computed in constant time, this property allows SVMs to find a non-linear classifier without increasing the complexity. Common kernel functions are high degree polynomials, radial basis functions, and sigmoid functions.

4.2.2. *Discriminating between manufacturing processes*

All mechanical parts in our study share some subset of curvature feature statistics (i.e., those resulting from material removal operations in machining); however, the casting procedure generates surfaces that result in a part being easily removable from the cast or mold. As described in Section 4.1.3, cast parts have tell-tale curvature values that signify the use of a cast-then-machined process, where machining features are used to "finish" the part. It is intuitive that applying a classifier that can be tuned to weight these attributes differently should perform better, and SVMs possess this attribute. Weights for the projected features are learned through an explicit training process on the SVMs. Distance-based classifiers, such as nearest neighbor and unsupervised machine learning by clustering the models, often consider equal weighting of attributes during distance computation.

SVMs are used to adapt the available features to the characteristic of this classification problem. The training process treats the curvature-based shape descriptor as an input vector on which the SVMs learn which regions of the vector (i.e., what ranges of curvature values) are most characteristic of the distinction between prismatic-machined and cast-then-machined CAD models. The overall approach can be summarized as

1. *Pre-processing phase*: compute the per-vertex curvature and construct the curvature-based shape descriptors for the training models.
2. *Training phase*:
   - (a) *Nearest neighbors*: no training is required.
   - (b) *SVMs*: train a classifier using SVMs with the classifications and corresponding example curvature-based shape descriptors.
3. *Query phase*: compute the per-vertex curvature and construct the shape descriptor for a query model.
4. *Classification phase*: classify the query model based on the similarity of its shape descriptor to those in the database:
   - (a) *Nearest neighbors*: return the classification of the nearest example model(s).
   - (b) *SVMs*: feed the query model shape descriptors into the trained SVMs classifier.

4.2.3. *Classification by nearest neighbors*

The common nearest neighbor approach is presented as a baseline for comparing the utility of classifying curvature statistics descriptors. Nearest neighbor classifier returns the classification of the query model's closest example model(s). Minkowski L_2, Euclidean, distance is used to determine the distance in between curvature shape descriptors. L_2 distance between models t and q is computed as

$$L_2(t,q) = \left(\sum \|t_i - q_i\|^2\right)^{1/2}.$$

The closest example model(s) of the query model determines the returned classification. The nearest neighbor classifier is included in the evaluation as an alternative to SVMs and provides a performance baseline for the different classifiers.

4.2.4. *Classification with SVMs*

The separation in between prismatic-machined and cast-then-machined processes can be learned by feeding example curvature shape descriptors and their corresponding categories to SVMs. The SVMs framework allows descriptors to be non-linearly projected into a high dimensional feature space for the algorithm to find a linear separating margin. However, this may result in a non-linear separating margin in the low-dimensional input space.

As described in Section 4.2.1, the learning algorithm takes advantage of L_D only depending on dot products of $\mathbf{x}_i \cdot \mathbf{x}_j$, rather than direct projection of examples to a high-dimensional feature space, $\mathbf{x} \rightarrow \Phi(\mathbf{x}_i)$. Hence, it replaces the dot product computation with a kernel function, $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, and that computes the dot products of the projected examples in constant time. This approach allows the SVMs to find non-linear classifications without increasing computational complexity.

Kernel functions need to satisfy the Mercer condition:

$K(\mathbf{x}_i, \mathbf{x}_j) > 0.$

This ensures that the matrix is positive semidefinite in the quadratic programming formulation. Therefore, a solution is guaranteed to exist.

We evaluated several different common kernel functions for SVMs learning:

- High degree polynomial: $(\gamma(\mathbf{x}_i \cdot \mathbf{x}_j) + r)^d, \ \gamma > 0$,
- Radial basis (Gaussian) function: $\mathrm{e}^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|}, \ \gamma > 0$,
- Sigmoid function: $\tanh(\gamma(\mathbf{x}_i \cdot \mathbf{x}_j) + r), \ \gamma > 0$,

where γ, r and d are free parameters of the kernel functions.

For the radial basis function kernel, N_s (i.e., the number of support vectors) Gaussian functions are centered at the selected support vectors. The weights and thresholds for each Gaussian are determined during the SVMs training. In contrast, the Sigmoid function models a specific kind of two layer neural network. The first layer consists of N_s sets of weights, each set consisting of D (i.e., the data dimension) weights. The second layer consists of N_s weights. Evaluation takes a weighted sum of Sigmoid functions.

The free parameters of the kernel functions and the penalty parameter C need to be determined by a model selection process. Our approach follows the advice of Lin [44] on performing an exhaustive grid search and cross validation for optimal parameters. Parameters are estimated by trying a growing sequence and cross validation divides training data into n folds. A classifier is trained on $n-1$ folds for classifying the remaining fold. Accuracies are averaged across different classifications to predict testing performance. Parameters with the highest cross validation performance are selected for training the final classifier.

5. Empirical evaluation

This section presents a set of empirical studies that show both the current state-of-the-art as well as the positive improvement of the Curvature and SVMs approach over existing techniques and the nearest neighbor classification approach. Different approaches were applied to learn and classify a dataset of over 100 hand classified prismatic-machined or cast-then-machined CAD models. Experiments were conducted using a set of the mechanical part data sampled from the National Design Repository. All models can be retrieved at http://www.designrepository.org/datasets/.

5.1. Results: process classification with existing techniques

Current research on shape matching techniques focuses on extracting features to match the gross shape of mesh models. For example, shape distributions sample distances in between surfaces; Zernike descriptors record the transformation of models from a sphere; reeb graph technique segments and matches shapes by their topology; and finally scalespace decomposition separates models by shape features.

An empirical assessment of existing techniques was done using the manufacturing process dataset introduced earlier in this paper. The results clearly show that existing techniques are not capable of adequately distinguishing artifacts manufactured by different processes. Table 2 shows the nearest neighbor classification accuracy for some mesh-based shape matching techniques. In these experiments, the classification of closest example model is assigned to the query model.

Why is processes classification a problem? Characteristics of the manufacturing processes show why current shape matching features fail to separate artifacts produced by the processes. Prismatic machining manufactures artifacts by material removal operations on regular shaped stock, such as rectangular blocks or cylinders. Cast-then-machined manufacturing first cast out stocks with the gross shape of the resulting parts and then lets machining operations construct the details.

The engineering rationale in this classification is that parts that are exclusively machined are usually high-precision parts or parts made in small batches (i.e., for custom jobs). Cast-then-machined parts are typically from larger production runs and generally have much looser tolerance considerations for the non-machined surfaces of the object. In the latter case the investment of the physical plant is larger, as is the manufacturing production plan (i.e., one needs to machine a mold with which to do casting).

Machining is a high precision and time-consuming process. It requires a process plan to route tools for

Table 2
Classification accuracies of shape descriptors

Shape matching techniques	Accuracy (%)
Shape distributions [16]	57
Scalespace [21]	54
Reeb graph [15]	55
Zernike moments [17]	52

material removal operations. The application of the casting procedure loosely constructs the shape of parts and reduces the need for machining; hence it decreases the complexity of process plans, manufacturing time and costs.

A similar setup of manufacturing equipment is able to produce a variety of artifacts in different shapes. Further, artifacts' shapes can be drastically different when the same set of volume removal operations is applied to different shapes of stock. The resulting shapes of the artifacts are no longer strongly tied with their respective manufacturing processes.

5.2. Curvature-based process classification

The curvature-based descriptor along with the nearest neighbor and SVMs classifiers were applied to learn and classify the sprismatic-machined or cast-then-machined CAD models dataset.

5.2.1. Experimental set up

The experiment was repeatedly performed on randomly selected training models to confirm the robustness of classification.

The classified manufacturing model dataset was split into random halves for training and testing. Each experiment was repeatedly performed for 40 times to confirm the robustness of the approach. The objective was to verify that the system could stably learn the classification by randomly selecting sets of training examples and then accurately classify the non-training parts. High fidelity mesh representations of the CAD models were prepared by faceting the ACIS SAT solid models using Geomagic Studio. On average 150,000 triangles were used to approximate the surfaces per solid model to ensure that the curvature computation is accurate.

Curvatures for every vertex on each training example were estimated to construct the curvature shape descriptor. The implementation of the curvature estimation algorithm is provided by `trimesh2`. Curvature histograms used in this evaluation consists of 100 bins. SVMs implementation was provided by `libsvm`. All experiments were performed on the Linux platform using a single 1.5 GHz AMD Opteron processor with 1 GB of memory.

The curvature estimation and SVMs learning procedure took 5 min to build a classifier from half of the labeled dataset. Constructing curvature shape descriptors and the subsequent classification of all the query models using the SVMs classifier took 5 s. Estimating per vertex principle curvatures and constructing curvature histograms took approximately 2 s per model with 100,000 triangles. The SVMs training process took 0.2–0.5 min, depending on the example model set. Most of the training time was spent on model selection for optimal parameters using grid searches and cross validations.

In the SVMs classification experiments, due to the limited size of the test dataset, about 80% of the training examples were selected as support vectors to define the separating margin during the training phase in each run of the experiment. Although there is still no theoretical relationship on how the number of support vectors would affect the classification performance, a large number of support vectors may indicate possible overfitting. To alleviate this problem, the experiment was repeatedly performed to validate the performance statistics. Average and maximum accuracies are provided to illustrate the performance. Higher maximum and average accuracies show a better classification rate. Lower standard deviation shows the classifier being more stable.

5.2.2. Results: curvature and nearest neighbor classification

The nearest neighbor classifier determines the classification of the query model by its nearest example neighbor, using the proposed curvature shape descriptor and Euclidean distance. Table 3 shows a summary of the results.

5.2.3. Results: curvature and SVMs-based classification

The SVMs classification of curvature shape descriptors was evaluated along with different non-linear kernel functions (radial basis function, polynomial, and sigmoid functions). κ_1, κ_2, mean curvature, H, and Gaussian curvature, K, histograms were computed for this evaluation. Table 4 shows a summary of the results.

Using minimum curvature, κ_2, features along with radial basis function SVMs produced the highest classification rate of 87% with an average of 77.1% and the lowest standard deviation of 4.75%. These statistics show this combination produced a more accurate classification with a higher stability. This is a 20–30% increase over the existing 3D shape matching algorithms (Table 2, accuracy 53–57%). κ_2 performed the best in this classification because most curved surfaces were generated by volume removal operations, such as holes, slots, and pockets, and these machining features have zero minimum curvature. Therefore, κ_2 facilitated classification by minimizing the variety of curvature statistics for prismatic-machined parts.

The results of the experiments show that the radial basis function performed slightly better than the polynomial or sigmoid kernel functions. The classification accuracies ($>75\%$) of minimum curvature were always higher than other curvature statistics despite which different kernel function was used. Maximum and mean curvature performed similarly with an average of 70% accuracy. Gaussian curvature performed the worst (average 60%) in

Table 3
Nearest neighbor and curvature statistics classification accuracy

	κ_1 (%)	κ_2 (%)	H (%)	K (%)
Maximum	69	55	59	58
Average	54.59	48.62	51.63	52.1
Standard deviation	6.66	3.89	5.19	4.62

Table 4
SVMs and curvature statistics classification accuracy

	κ_1 (%)	κ_2 (%)	H (%)	K (%)
(a) *Polynomial Kernel function*				
Maximum	81	**87**	79	74
Average	69.7	75.5	70.95	65.2
Standard deviation	6.57	4.81	5.39	6.12
(b) *Radial basis Kernel function*				
Maximum	85	**87**	**87**	76
Average	67.85	**77.1**	71.15	59.2
Standard deviation	8.13	**4.75**	7.9	6.7
(c) *Sigmoid Kernel function*				
Maximum	**87**	85	81	72
Average	69.7	75	67.5	59.35
Standard deviation	7.41	5.87	7.24	7.28

Bold numbers highlight the highest accuracies and lowest standard durations.

the experiments. This shows that the Gaussian curvature statistics did not properly separate the CAD models according to the prismatic-machined and cast-then-machined manufacturing classifications.

5.3. Discussion

The nearest neighbor classification did not perform satisfactorily with curvature shape descriptors. Only 50–60% of models were classified correctly for each run of the experiment. In contrast, the radial basis function SVMs and minimum curvature performed 20–30% better. This result demonstrates that the SVMs classifiers found significantly better separation among the curvature shape descriptors with respect to the targeted classifications. This experiment also shows the choice of classifiers could considerably affect the performance in CAD classification. This does not mean that SVMs are always superior to nearest neighbor classifier. It is important to search for a combination of features and classifiers that produces the best accuracy.

Even though this test was done with only two classes of models, the technique can be generalized to multiple classifications. Nearest neighbor classifier naturally handles multiple classes and the SVMs classifier has been extended to handle multiple classes [45,46]. The issue in generalization is that the community of users needs to develop the additional feature extractors to discriminate the other example classifications of interest. Further, it may be the case that some kinds of classifications and discriminations are simply not feasible from low-level shape information (i.e., objects with very different functions may have similar gross shapes).

6. Conclusions

This paper described a new approach to automate the classification of 3D mesh-based representations of mechanical CAD models according to manufacturing processes. The contribution of this research is the introduction of a new shape descriptor based on surface curvature and use of the SVMs to learn the manufacturing categorization of CAD models. This approach relates automatic mesh model classification to a practical application and can support relevant engineering interrogations such as cost estimation and variational design. In addition, the paper introduced a dataset for use in comparing the performance of 3D search techniques in the domain of 3D mechanical CAD models and a detailed empirical assessment of the manufacturing process classification technique. A contribution of this research is the establishment of additional datasets for evaluating model retrieval techniques on CAD/CAM artifacts.

In developing the classifier for prismatic-machined and cast-then-machined manufacturing processes, we have shown that the combination of a curvature-based descriptor and non-linear SVMs demonstrates a considerable improvement over recent shape descriptors and nearest neighbor classification. However, it is also evident that new engineering-relevant features and classifiers are needed for the challenging domain of CAD model databases. The need specifically exists for a greater variety of low-level feature descriptors, especially ones that can capture non-local shape configurations (i.e., slots, pockets, or other features that are not adjacent to each other on the surface of the artifact). With additional shape analysis features and different classification schemes available, a future goal is to apply supervised machine learning to automatically select relevant features, resulting in adaptive matching systems.

It is the belief of the authors that manufacturing and engineering-relevant classifications are an open challenge problem for the 3D retrieval community. Results [30] have shown that most current shape retrieval techniques perform unacceptably when asked to classify objects as cast or prismatically machined parts. Readers may feel that this distinction is too subtle, but in actuality this distinction is readily identifiable in the micro-geometry of the objects themselves. Further, this a binary classification—the simplest possible. Considerable research needs to be performed before classifiers will be able to distinguish among objects across a wider variety of manufacturing processes as well as answer more complex engineering queries.

Acknowledgments

This work was supported by the Library of Congress' National Digital Information Infrastructure and Preservation Program (NDIIPP) through the National Science Foundation (NSF) DIGARCH Program Grant IIS-0456001. Additional support includes NSF CAREER Award CISE/IIS-9733545 and Office of Naval Research (ONR) Grant N00014-01-1-0618.

Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the other supporting government and corporate organizations.

References

[1] Iyer N, Jayanti S, Lou K, Kalyanaraman Y, Ramani K. Three dimensional shape searching: state-of-the-art review and future trends. Computer Aided Design 2005;37(5):509–30.

[2] Cardone A, Gupta SK, Karnik M. A survey of shape similarity assessment algorithms for product design and manufacturing applications. Journal of Computing and Information Science in Engineering 2003;3:109–18.

[3] Veltkamp RC, Tangelder JW. A survey of content based 3d shape retrieval methods. In: Shape modeling and applications; 2004.

[4] Snead CS. Group technology: foundations for competitive manufacturing. New York: Van Nostrand Reinhold; 1989.

[5] Han J-H, Regli WC, Pratt MJ. Algorithms for feature recognition from solid models: a status report. IEEE Transactions on Robotics and Automation 2000;16(6):782–96.

[6] Shah J, Anderson D, Kim YS, Joshi S. A discourse on geometric feature recognition from cad models. ASME Transactions, the Journal of Computer and Information Science in Engineering 2001;1(1):41–51.

[7] Ji Q, Marefat MM. Machine interpretation of cad data for manufacturing applications. Computing Surveys 1997;29(3):264–311.

[8] Kyprianou LK. Shape classification in computer aided design. PhD thesis, Christ College, Cambridge, UK, University of Cambridge; July 1980.

[9] Elinson A, Nau DS, Regli WC. Feature-based similarity assessment of solid models. In: Hoffman C, Bronsvoort W, editors. Fourth symposium on solid modeling and applications, New York, NY, USA. Atlanta, GA: ACM Press; 1997. p. 297–310.

[10] Cicirello V, Regli WC. Resolving non-uniqueness in design feature histories. In: Anderson D, Bronsvoort W, editors. Fifth symposium on solid modeling and applications, New York, NY. Ann Arbor, MI, USA: ACM Press; 1999.

[11] Cicirello V, Regli WC. An approach to a feature-based comparison of solid models of machined parts. Analysis, and Manufacturing (AIEDAM) 2002;16(5):385–99.

[12] McWherter D, Peabody M, Shokoufandeh A, Regli W. Solid model databases: techniques and empirical results, ASME/ACM Transactions. The Journal of Computer and Information Science in Engineering 2001;1(4):300–10.

[13] Cardone A, Gupta SK, Karnik M. Identifying similar parts for assisting cost estimation of prismatic machined parts. In: ASME DETC design for manufacturing conference; 2004.

[14] Thompson WB, Owen J, de St. Germain H, Stark Jr S, Henderson T. Feature-based reverse engineering of mechanical parts. IEEE Transactions on Robotics and Automation 1999; 12(1):57–66.

[15] Hilaga M, Shinagawa Y, Kohmura T, Kunii TL. Topology matching for fully automatic similarity estimation of 3d shapes. In: SIGGRAPH. New York, NY, USA: ACM Press; 2001. p. 203–12.

[16] Osada R, Funkhouser T, Chazelle B, Dobkin D. Shape distributions. ACM Transactions on Graphics 2002;21(4):807–32.

[17] Novotni M, Klein R. Shape retrieval using 3d zernike descriptors. Computer Aided Design 2004;36(11):1047–62.

[18] Kazhdan M, Funkhouser T, Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3d shape descriptors. In: Eurographics/ACM SIGGRAPH symposium on Geometry processing; 2003. p. 156–64.

[19] Ip CY, Lapadat D, Sieger L, Regli WC. Using shape distributions to compare solid models. In: Seventh ACM symposium on solid modeling and applications; 2002.

[20] Ip CY, Sieger L, Regli WC, Shokoufandeh A. Automated learning of model classifications. In: Eighth ACM symposium on solid modeling and applications; 2003. p. 322–7.

[21] Bespalov D, Shokoufandeh A, Regli WC, Sun W. Scale-space representation and classification of 3d models, ASME Transactions. Journal of Computing and Information Science in Engineering 2003;3:315–24.

[22] Iyer N, Jayanti S, Lou K, Kalyanaraman Y, Ramani K. Shape-based searching for product lifecycle applications. Computer Aided Design 2005;37(13):1435–46.

[23] Iyer N, Kalyanaraman Y, Lou K, Jayanti S, Ramani K. A reconfigurable, intelligent 3d engineering shape search system part I: shape representation. In: ASME DETC 2003 computers and information in engineering conference; 2003.

[24] Iyer N, Kalyanaraman Y, Lou K, Jayanti S, Ramani K. A reconfigurable, intelligent 3d engineering shape search system part II: database indexing, retrieval and clustering. In: ASME DETC 2003 computers and information in engineering conference; 2003.

[25] Pal P, Tigga A, Kumar A. Feature extraction from large cad databases using genetic algorithm. Computer Aided Design 2005;37(5):545–58.

[26] Nene SA, Nayar SK, Murase H. Columbia object image library (coil-100). Technical Report, Columbia University; February 1996.

[27] Shilane P, Min P, Kazhdan M, Funkhouser T. The princeton shape benchmark. In: Shape modeling and applications; 2004. p. 167–80.

[28] Regli WC, Gaines DM. An overview of the NIST repository for design, process planning, and assembly. Computer Aided Design 1997;29(12):895–905.

[29] Regli WC, Cicirello V. Managing digital libraries for computer-aided design. Computer Aided Design 2000;32(2):119–32 special issue on CAD after 2000. Mohsen Rezayat, Guest, editor.

[30] Bespalov D, Ip CY, Regli WC, Shaffer J. Benchmarking search techniques for cad. In: 2005 ACM symposium on solid and physical modeling; 2005.

[31] Iyer N, Jayanti S, Ramani K. An engineering shape benchmark for 3D models. In: Proceedings of ASME IDETC/CIE; 2005.

[32] Smith SDG, Escobedo R, Anderson M, Caudell TP. A deployed engineering design retrieval system using neural networks. IEEE Transactions on Neural Networks 1997;8(4):847–51.

[33] Hou S, Lou K, Ramani K. Svm-based semantic clustering and retrieval of a 3d model database. Computer Aided Design and Application 2005;2:155–64.

[34] Elad M, Tal A, Ar S. Content based retrieval of vrml objects—an iterative and interactive approach. In: Eurographics Multimedia Workshop; 2001.

[35] Taubin G. Estimating the tensor of curvature of a surface from a polyhedral approximation. In: ICCV95; 1995. p. 902–7.

[36] Page DL, Sun Y, Koschan AF, Paik J, Abidi MA. Normal vector voting: crease detection and curvature estimation on large, noisy meshes. Graphical Models 2002;64:199–229.

[37] Meyer M, Desbrun M, Schröder P, Barr AH. Discrete differential-geometry operators for triangulated 2-manifolds. In: VisMath; 2002.

[38] Goldfeather J, Interrante V. A novel cubic-order algorithm for approximating principal direction vectors. ACM Transaction on Graphics 2004;23(1):45–63.

[39] Guillaume Lavoué FD, Baskurt A. A new cad mesh segmentation method, based on curvature tensor analysis. Computer Aided Design 2005;37(10):975–87.

[40] Rusinkiewicz S. Estimating curvatures and their derivatives on triangle meshes. In: Second international symposium on 3D data processing, visualization, and transmission. IEEE Computer Society; 2004. p. 486–93.

[41] Tangelder JW, Veltkamp RC. Polyhedral model comparison using weighted point sets. International Journal of Image and Graphics 2003;3(1):209–29.

[42] Cortes C, Vapnik V. Support vector networks. Machine Learning 1995;20(3):273–97.

[43] Burges CJC. A tutorial on support vector machine for pattern recognition. Data Mining and Knowledge Discovery 1998;2(2): 121–67.

[44] Hsu C-W, Chang C-C, Lin C-J. A practical guide to support vector classification, available at ⟨http://www.csie.ntu.edu.tw/cjlin/libsvm⟩; 2004.

[45] Weston J, Watkins C. Support vector machines for multiclass pattern recognition. In: Proceedings of the seventh European symposium on artificial neural networks; 1999.

[46] Hsu C-W, Lin C-J. A comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 2002;13(2): 415–25.