3D CAD model matching from 2D local invariant features

Article *in* Computers in Industry · June 2010

DOI: 10.1016/j.compind.2009.11.001 · Source: DBLP

4 authors, including:

Kunpeng Zhu
Hefei Institute of Physical Sciences, Chinese Academy of Sciences
46 PUBLICATIONS **724** CITATIONS

SEE PROFILE

Computers in Industry 61 (2010) 432–439

Contents lists available at ScienceDirect

Computers in Industry

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

3D CAD model matching from 2D local invariant features

K.P. Zhu *, Y.S. Wong, W.F. Lu, H.T. Loh

Lab of Concurrent Engineering and Logistics, Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore

ARTICLE INFO

Article history:
Received 25 September 2008
Received in revised form 19 September 2009
Accepted 5 November 2009
Available online 5 December 2009

Keywords:
CAD models
PCA
SIFT
Model matching

ABSTRACT

The matching of particular types of CAD models to existing physical models can provide invaluable support to the process of CAD design and reuse. To meet the demand for fast and robust algorithms to detect predefined models in database, an local invariant model matching approach is proposed in this paper. It first maps the 3D CAD model to 2D principal image plane by its first two principal components, and then finds affine invariant key points in the 2D image. The CAD model matching problem is implemented as key points matching. Experimental results show the proposed 3D model retrieval method performs fairly well in retrieving similar models from a database of 3D CAD models.

1. Introduction

With the common availability of 3D CAD databases in industry, a productive approach is to reuse and modify similar existing models instead of always creating new ones. The development of efficient search mechanisms is necessary to speedily identify and retrieve 3D models from large repositories. There are two basic types of approaches for the matching and retrieval of 3D CAD data: manufacturing feature-based techniques and shape content-based techniques.

Early works of CAD model matching focus on manufacturing feature-based techniques [1–3]. Manufacturing feature-based techniques extract manufacturing-related features based on the application of Group Technology (GT) applied to mechanical parts as index for retrieval [2,3]. Elinson et al. [1] used feature-based reasoning for the retrieval of solid models for variant process planning. Cicirello and Regli [4] examined issues pertaining to graph-based data structures and proposed a model dependency graph (MDG) approach to determine the machining features. Mcwherter et al. [5] have integrated the ideas in [1,2,4] with database techniques to enable indexing and clustering of CAD models based on shape and topology properties. Cardone et al. [6] compared machining features of prismatic machined parts for manufacturing cost estimation. However, the GT coding of a given CAD model is not unique, and it is difficult to obtain consistent manufacturing features. In fact, the issue of multiple feature interaction is a common problem to all existing manufacturing feature-based approaches [7]. The non-uniqueness of feature descriptions in different interactions has been one of the major hindrances in feature extraction from CAD solid models. Although Bayesian Networks (BN) [8] and MDG approaches [4] have been proposed and can successfully extract the interacting features for many part shapes, it is difficult to generalize these methods to all shape types due to their constraints on the surface topology and limited machining features included.

Recently, there are more reported works on shape content-based techniques for model matching. With the contributions from computer vision and pattern recognition, the content-based methods do not require any annotation as those for textual search. They only require robust 3D shape feature extraction that can be applied automatically. In these methods, content features can be extracted from polygonal mesh models for the determination of similarity among 3D models in the matching stage. Most techniques are developed for 3D multimedia model retrieval, such as those using shape distribution [9] spherical harmonics [10], light filed [11] and spectral approaches [12]. A comprehensive survey on shape content-based retrieval methods is given by Bustos et al. [13]. These techniques are recently applied to compare CAD models. Ip et al. [14] compare 3D CAD models using shape distributions and Bespalov et al. with hierarchy scale-space representations [15]. Ip and Regli [16] introduced a curvature-based shape descriptor with support vector machines (SVM) for manufacturing process discrimination. Statistics on surface curvatures are extracted for SVMs to learn as a separator for different shape models. Hou et al. [17] use shape information to cluster the semantics of parts by SVMs to learn 3D shapes for a system to recognize 3D CAD models. This study was extended to a 2D sketch-based part retrieval approach [18] and an integrated

* Corresponding author. Tel.: +65 65164857.
E-mail address: mpezhuk@nus.edu.sg (K.P. Zhu).

0166-3615/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2009.11.001

retrieval system of 2D sketches and 3D models [19]. For the 2D sketches, the alignment of the three orthogonal views sketches is a problem, and misalignment may lead to different features and poor matching. In the CAD-oriented search system, Iyer et al. [20,21] include most of the content-based approaches for 3D multimedia shapes. Pertaining to these methods, recent research efforts in matching of engineering parts are given in a survey by Cardone et al. [7] and Iyer et al. [22]. In the above approaches for 3D CAD model retrieval, with exception of [18,19], the features are computed from 3D surfaces, which is computationally expensive and time consuming, especially when the models are large.

A more suitable approach is to find the features from lower dimension attributes, while keeping most of the information of the 3D CAD model. There are various dimension reduction approaches studied lately, such as isomap [23], kernel approach [24], and local linear embedding [25], diffusion maps [26], Laplacian eigenmaps [27], etc. The review on dimensionality reduction techniques by Maaten et al. [28] shows that these techniques perform well on simulated data, but do not outperform the traditional Principal Component Analysis (PCA) approach [29] in real models. The PCA approach is simple, stable and computationally fast in the study. On the other hand, it could be useful for extraction of distinctive regional features over different scales, rotations, and locations for the matching tasks. Recent surveys have shown that the scale invariant features transform (SIFT) technique by Lowe [30,31] is the most efficient for several criteria [32]. In this paper, we propose a 3D shape representation scheme based on the combination of 3D model dimension reduction with PCA analysis and 2D image matching through SIFT. The proposed 3D shape representation scheme consists of two steps. The section below describes the transformation of a 3D model to a 2D image by projecting the vertices of the model onto its principal plane with PCA, and presents the proposed retrieval strategy, including the extraction of a CAD model for a feature database and classification with a robust k-nearest neighbor (k-NN) strategy. The following section shows the experimental results and compares those with both 3D and 2D feature extraction approaches.

2. Dimension reduction of 3D model and SIFT image features

The goal is to provide a method based on 2D images for fast computation and matching of 3D models. The extracted features for matching are rotation, scaling and translation (RST) invariant. Fig. 1 shows an overview of the proposed approach. It consists of three steps. The 3D model is first mapped to its 2D principal plane and then the RST features are extracted in the plane by approaches discussed in Section 2.2. The extracted features then form a feature database for the indexing of the 3D database $\mathrm{SIFT_{DB}}$ with scale invariant features transform. When a query 3D model is given, the RST invariant features of its corresponding principal plane are extracted as $\mathrm{SIFT_{IN}}$ and used to match those in the database ($\mathrm{SIFT_{DB}}$). The retrieval results are then ranked by the similarity measurement.

2.1. 2D principal plane of 3D CAD model

Basically in PCA [29], the aim is to find components $S_1, S_2, \ldots, S_n$ that manifest the maximum amount of variance possible by n linearly transformed components. The direction of the first principal component w_1 is defined by

$$w_1 = \arg\max E\{w^T x)^2\}, \quad |||w|| = 1, k = 1, 2, \ldots n \tag{1}$$

where w_1 is of the same dimension m as the random data vector x. Thus the first principal component is the projection along the direction in which the variance of the projection is maximized. Having determined the first $k-1$ principal components, the kth principal component is determined as the principal component of the residual:

$$w_k = \arg\max E\{[w^T (x - \sum_{i=1}^{k} w_i w_i^T x)\ ]^2\} \tag{2}$$

The principal components are then given by $s_i = w_i^T x$. In practice, the computation of w_i can be simply accomplished using the (sample) covariance matrix $C = E\{xx^T\}$, i.e. the set of w_i constitutes the eigenvectors of C corresponding to the n largest eigenvalues of C.

Fig. 2 illustrates the process to get the PCA 2D image (principal image) of a 3D model. The solid model (in STL format) is first converted to its 3D mesh model for further processing. As the focus is on shape matching, the color attribute of the model is not considered in the mesh model. With the mesh model, PCA is applied to the vertex while the topology of face properties is retained. As a result we get the image of the 3D model in its principal plane as shown in Fig. 2c. The principal image is similar to a perspective projection, where we can see the inner structure of the 3D model, while its top view (Fig. 2d) and side view (Fig. 2e) cannot show these features. Hence, the principal image can capture much of the 3D information and the inner structure of the 3D model.

It is noteworthy that for these geometric images, Laplacian [33] is typically applied to smooth the image before further processing. It is not necessary in this approach however, as the next section will show that the smoothing is achieved in the scale invariant feature transforms (SIFT) at larger scales.

2.2. Scale invariant feature transforms (SIFT) on the principal plane

After extracting the principal image of the 3D model, the next step is to extract its RST invariant features to represent the model. In recent years, local image descriptors have received much attention in computer vision tasks, such as image retrieval, image comparison, and feature matching for 3D reconstruction. Schmid and Mohr [34] showed that local invariant feature matching could be extended to general image recognition problems for matching a feature against a large database of images. A comparison of such approaches is given in [32]. One of the most promising of these methods is the scale invariant feature transform (SIFT) developed by Lowe [31]. Lowe [30,31] developed SIFT features for extracting distinctive scale and orientation invariant features from images suitable for reliable matching. The following section briefly outlines the SIFT and the method used to match features. The processes are achieved via two stages, as SIFT detector, and as SIFT descriptor.

2.2.1. SIFT detector

The SIFT detector attempts to identify locations and scales that are identifiable from different views of the same object. This can be efficiently achieved using a "scale space" function:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{3}$$

where * is the convolution operator, $D(x, y, \sigma)$ is a Gaussian kernel $g(x, y, \sigma) = (1/(\sigma\sqrt{2\pi}))e^{-(x^2+y^2)/2\sigma^2}$ and $I(x, y)$ is the input image, where (x, y) is the spatial location and σ is the scale.

Difference of Gaussians (DoG) is then applied to locate scale-space extrema, $D(x, y, \sigma)$ by computing the difference between two images, one at scale of k times the other:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{4}$$

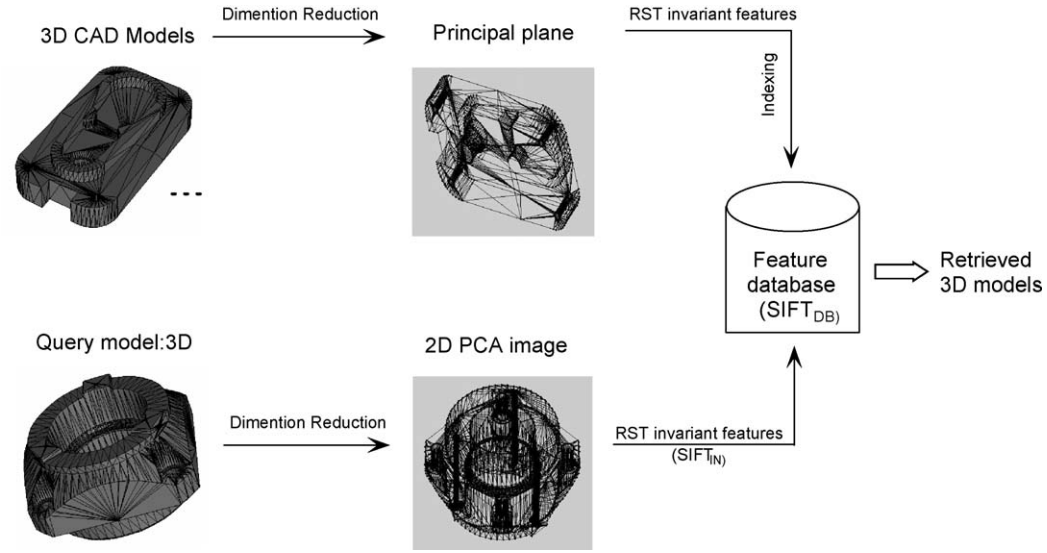

Fig. 1. Overview of 2D feature matching for the 3D model retrieval.

After each octave the image is down-sampled by a factor of 2. From these slices of scale space the DoG function is calculated by simply subtracting adjacent images in the stack. A local extremum of $D(x, y, \sigma)$ is then found by comparing each pixel to its 26 neighbors, with 8 at the same scale and two sets of 9 at the scales immediately above and below, respectively. If the resultant value is the minimum or maximum of all of the other points, then the point is an extremum. Each feature point is then assigned an orientation and a scale. The orientation is determined by constructing a histogram of gradient orientations in a neighborhood around the feature point [30]. The most dominant orientation is chosen to represent the orientation of the feature. The scale of the feature point corresponds to the scale at which it was found. This is a little similar to the edge detection with wavelet analysis [35] as blob-like structures are finally extracted across multiscales.

Fig. 3 demonstrates the SIFT feature detection. After the DoG analysis of the image at different scales (Fig. 3a), the maxima across the scales are computed and defined to be the key points. To find the local maxima, the image is smoothed several times with a Gaussian convolution mask at different scales. These smoothed versions are combined pair wise to compute a set of DoGs. After a few smoothing steps, the image can be sub-sampled to process the next octave. The DoG features detected in images given in our example are shown in Fig. 3c. Points with square are the remaining key points after removal of low contrast points and edge responses.

2.2.2. SIFT key point descriptor

Since every feature has an associated orientation and scale, a descriptor for each feature can be constructed by rotating and scaling the local neighborhood of each feature into a canonical view, and thereby lead to scale and rotation invariance. A key point descriptor typically uses a set of 16 histograms, aligned in a 4×4 grid, each with 8 orientation bins, one for each of the main compass directions, and one for each of the mid-points of these directions. It results in a feature vector containing 128 elements.

Fig. 4 illustrates this descriptor. The neighborhood around the feature is divided into a grid of 4×4 blocks and a gradient orientation histogram of each 4×4 block is constructed. Each

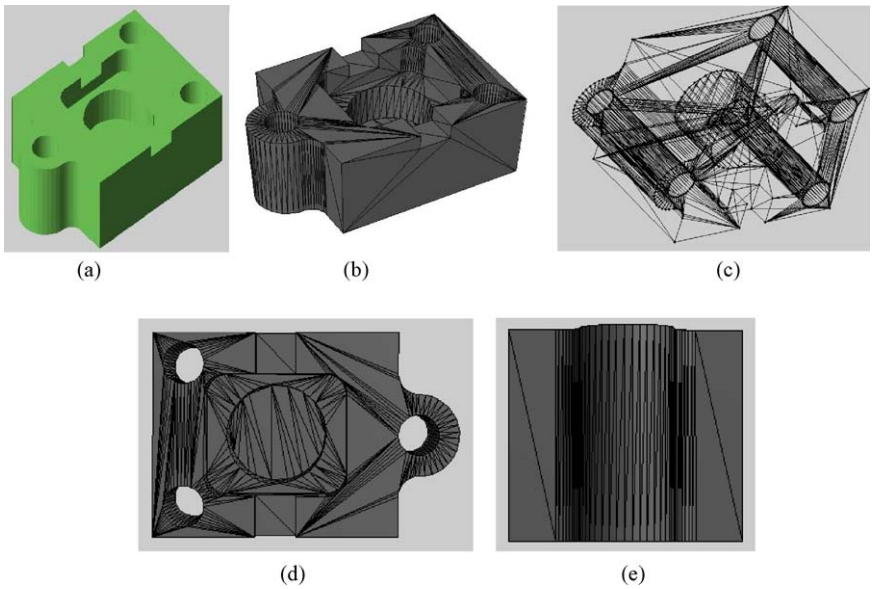

Fig. 2. Transformations of the 3D CAD model to PCA 2D image. (a) 3D CAD model, (b) 3D mesh model, (c) PCA 2D image, (d) top view and (e) side view.

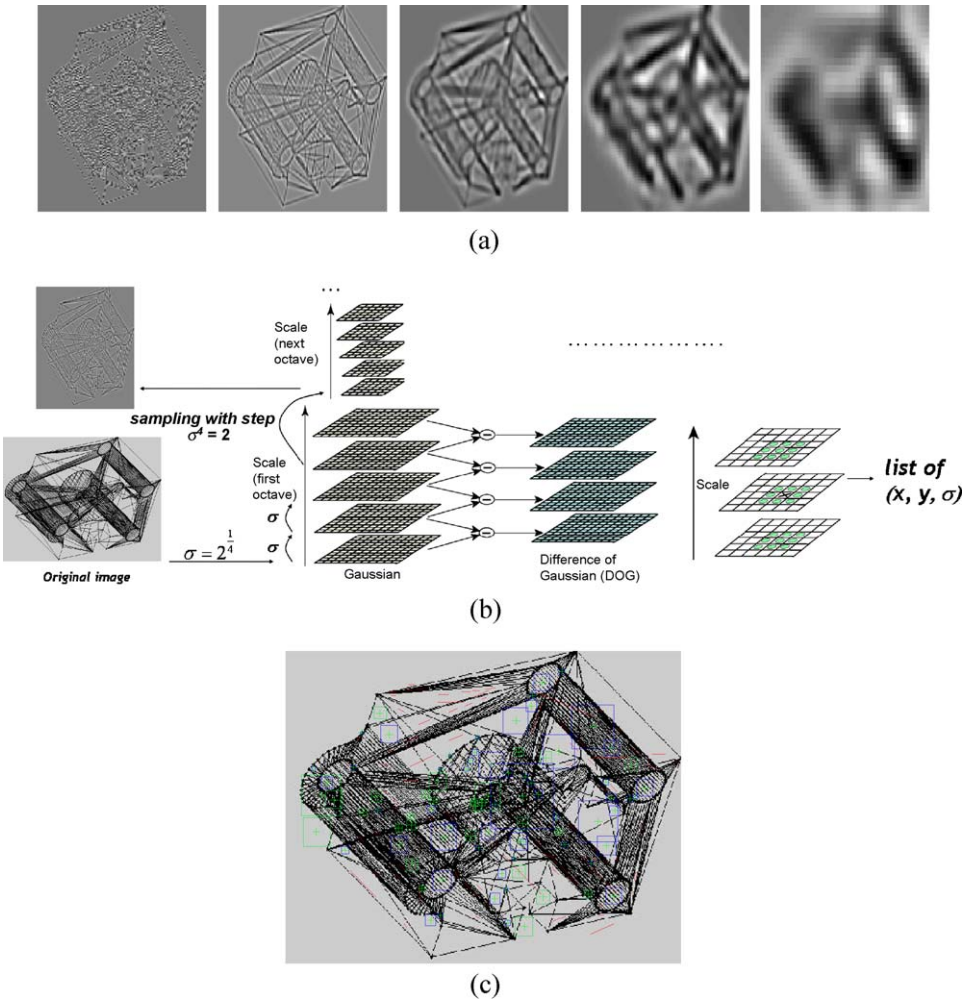

Fig. 3. SIFT key point detection from the 2D image. (a) DoG images at different scales, with scales increasing from left to right. (b) Overview of the DoG detection scheme. (c) Maxima of DoG across scales.

sample of the histogram is weighted by the gradient magnitude and a circular Gaussian window function. The local gradient data are also used to create key point descriptors. The gradient information is rotated to line up with the orientation of the key point and then weighted by a Gaussian with variance of 1.5 × key point scale [31]. This set of data is then used to create a set of histograms over a window centered on the key point.

The box on the left of Fig. 4 shows the local neighborhood of the feature with the gradients in each pixel. The red circle indicates a Gaussian window function. The right-hand box shows the resulting 2 × 2 set of orientation histograms with each gradient weighted by its magnitude and the window function. The resulting vectors are known as SIFT keys and are used in a *k*-nearest neighbor (*k*-NN) approach to identify possible objects in an image.

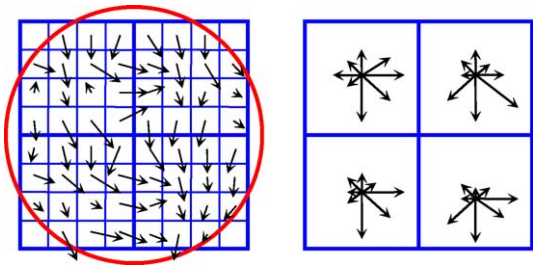

Fig. 4. Illustration of the SIFT feature descriptor.

2.3. *k*-Nearest neighbor (*k*-NN) classification

The *k*-NN is a pattern recognition algorithm [29] for classifying objects based on closest training examples in the feature space. An object is classified by a majority vote of its neighbors, with the object most common amongst its *k* nearest neighbors being assigned to the class. Collections of keys that agree on a possible model are identified. The matching of descriptors from an input image to the detector's database is done simply by finding the closest neighbor (3-nearest neighbors for example) in the database to the input descriptor. This is done in terms of Euclidian distance in this 128-dimensional descriptor space. To increase the recognition robustness, a generalized Hough transform [31] is then used to find clusters of feature points that correspond to those in the training images.

3. Experimental results

3.1. Mechanical CAD model dataset

Experimental results are presented to demonstrate the accuracy and performance of 3D CAD model retrieval from a dataset based on the proposed approach. The test dataset consists of mechanical CAD models available from the Purdue University's ESB database [36]. The dataset was functionally classified into three clusters of different types of basic geometric shapes obtained

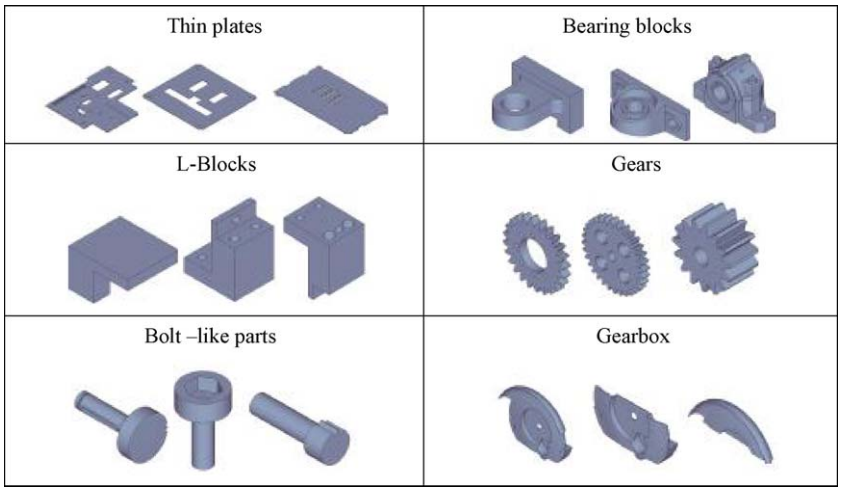

Fig. 5. Examples of models from ESB dataset [36].

from the ESB database: flat thin-wall component, rectangular-cubic prism, and solid of revolution. Training examples for weight learning were randomly chosen and *k*-NN classification was typically performed with $k = 3$ [31]. The system was able to classify all of the models that were not among those used in the training. Each experiment was repeated to verify the robustness of the proposed approach. Fig. 5 shows mechanical artifacts classified by their shapes. The names of the shape categories serve to reflect primitive functional characteristics of the models.

3.2. SIFT model matching

The 3D model matching based on SIFT features is divided into two distinct steps. The first step is achieved by finding scale-space extrema in the difference of Gaussian (DoG) pyramid. A point is taken as an extremum if it is below or above its 8 neighbors at the same scale and the 9 neighbors at higher and lower scales as earlier illustrated in Fig. 3b. A point of interest is then found at a given scale. Its major orientation is computed as the major direction of a patch of pixels around its position. The descriptor vector is next computed at the feature scale (Fig. 4). This is a vector of 16 histograms of gradient where each histogram contains 8 bins, leading to a 128-dimensional descriptor. Due to their construction, SIFT vectors are invariant by scale change and rotation.

As a pre-process step, SIFT vectors are computed from the input model ($SIFT_{IN}$ in Fig. 1). All descriptors of the database models have also been extracted offline ($SIFT_{DB}$, where features of the *i*th database model are shown as $SIFT^i_{DB}$). The first step is to identify correspondences between $SIFT_{IN}$ and $SIFT_{DB}$. Each feature from $SIFT_{IN}$ is compared to each feature from $SIFT_{DB}$ and only the matches whose distance is lower than a threshold ε are kept. For a query feature q of $SIFT_{IN}$, this can be seen as finding the ε-neighborhood of q in $SIFT_{DB}$. Once we have obtained for each descriptor of $SIFT_{IN}$ a list of its neighbors in $SIFT_{DB}$, these matches are fitted to SIFT transformations.

For the *i*th model in the database, we consider correspondences between $SIFT_{IN}$ and $SIFT^i_{DB}$. Each correspondence c between feature f_1 in $SIFT_{IN}$ and feature f_2 in $SIFT^i_{DB}$ can be seen as a four-dimensional point: $c\ (\theta, \eta, x, y)$ where θ is the rotation between the orientations of f_1 and f_2, η is the scale ratio between f_1 and f_2, and (x, y) is the coordinates of f_1 in the input image. Each correspondence is projected in a 4D grid. To reduce sensitivity to

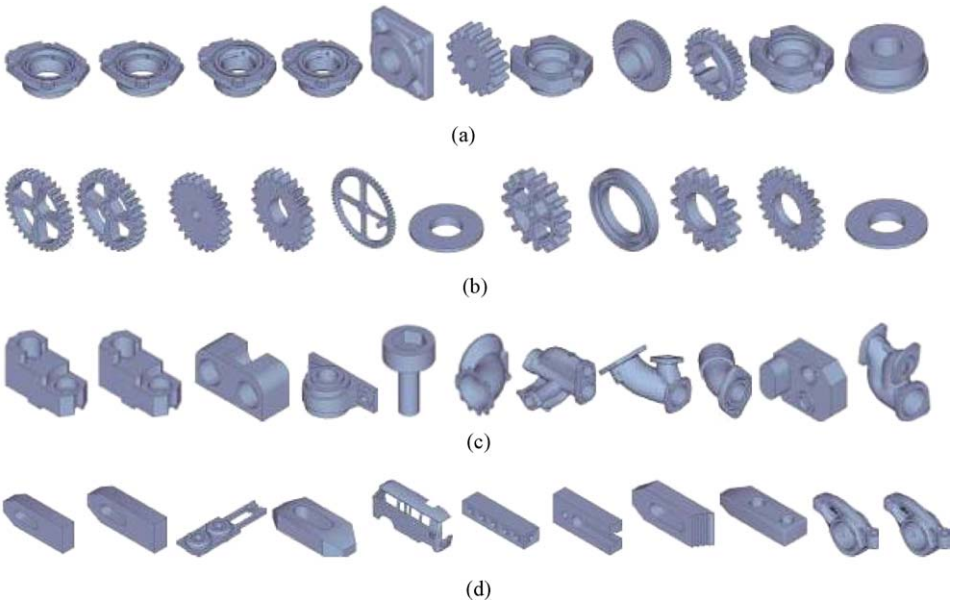

Fig. 6. Retrieval results of different shapes based on SIFT. (a) Oil pans, (b) gears, (c) parts and (d) parts.

the grid tile sizes, each point is projected on its two closest tiles on each dimension. Thus, a match is projected in 16 tiles. Eventually, every cluster with at least three correspondences can be fitted by a SIFT transformation and can be seen as a potential match.

3.3. Matching efficiency measurement

Apart from the classification performance, the efficiency of the proposed shape comparison method has also been evaluated in terms of information retrieval performance. In this case, each model of the database is used as query, and the retrieved shapes are ranked in terms of shape similarity to the query. The performances of various techniques are evaluated by the k-nearest neighbor classification (k-NN), and information retrieval precision-recall (IRPR) curve is used for evaluating the information retrieval systems [37].

Let Σ be the database of 3D shapes, and $\Sigma_i \ldots \Sigma_l$ be the disjunctive classified similar shapes, that is:

$$\Sigma_i = \{\Sigma_1^i, \ldots \Sigma_{n_i}^i\} \quad \Sigma_l^i \in \Sigma (1 \le l \le n_i), \text{ and if } i \ne j, \Sigma_i \cap \Sigma_j = \varnothing \tag{5}$$

Models Σ_1^i and $\Sigma_{n_i}^i$ are said to be relevant if they are from the same class Σ_i. The precision is the ratio of relevant matches m (from the same class as the query model) to the number of retrieved models r, and recall is the ratio of relevant matches m to the size n_i of query class Σ_i. More specifically, precision and recall are defined as:

$$\text{precision} = \frac{m}{r} \tag{6}$$

$$\text{recall} = \frac{m}{n_i - 1} \tag{7}$$

The k-NN classification labels a query model with the categories of its k closest neighbors, where k is the threshold for classification. The numbers of labeled categories potentially increase and decrease with respect to k. For each k, the mean of the recall and precision among all models was used as a representative value. To illustrate the results, the IRPR is plotted against recall on different datasets and comparison techniques. Ideally, a retrieval system should retrieve as many relevant models as possible, that is, both high precision as well as high recall are desirable. The formulas (6) and (7) show the trade-off between precision and recall. Trying to increase recall, typically, introduces more irrelevant models into the retrieved set, thereby reducing precision.

3.4. Results and discussions

The results are based on shape retrieval from the ESB database [36]. Fig. 6 shows the top 10 retrieval results for four cases obtained using the proposed approach. The first figure in each row is the query model, while the other 10 models (to the right) are the top 10 models ranked by the k-NN value. Fig. 6 depicts the corresponding IRPR curve for all geometrical descriptor vectors used. The query models in Fig. 6 represent different kinds of geometric structures. The first two are from solid of revolution clusters, and the query model 3 is from flat thin-wall component cluster and query model 4 is from rectangular-cubic prism cluster. The retrieval results are shown in Fig. 6a–d. It can be observed that the proposed approach can accurately retrieve all the closely matched models by ranking them top.

To demonstrate the effectiveness of our approach, we compare the results with those by the shape distribution [9] and spherical harmonics [10] approach which has been developed recently and attracted increasing interests on their application to 3D model matching. Besides, as one of the most typical and popular 2D image matching approaches, Harris–Laplacian affine invariants [38] is also implemented and compared to the proposed approach. The resultant precision (PR) curves of all the approaches are compared in Fig. 7 (where D2 represents shape distribution approach, while Harris represents the Harris–Laplacian approach, and SH represents the spherical harmonics approach.). As can be drawn from Fig. 7, the SIFT approach has consistently higher precision than the Harris, SH and D2 approaches for all the queries.

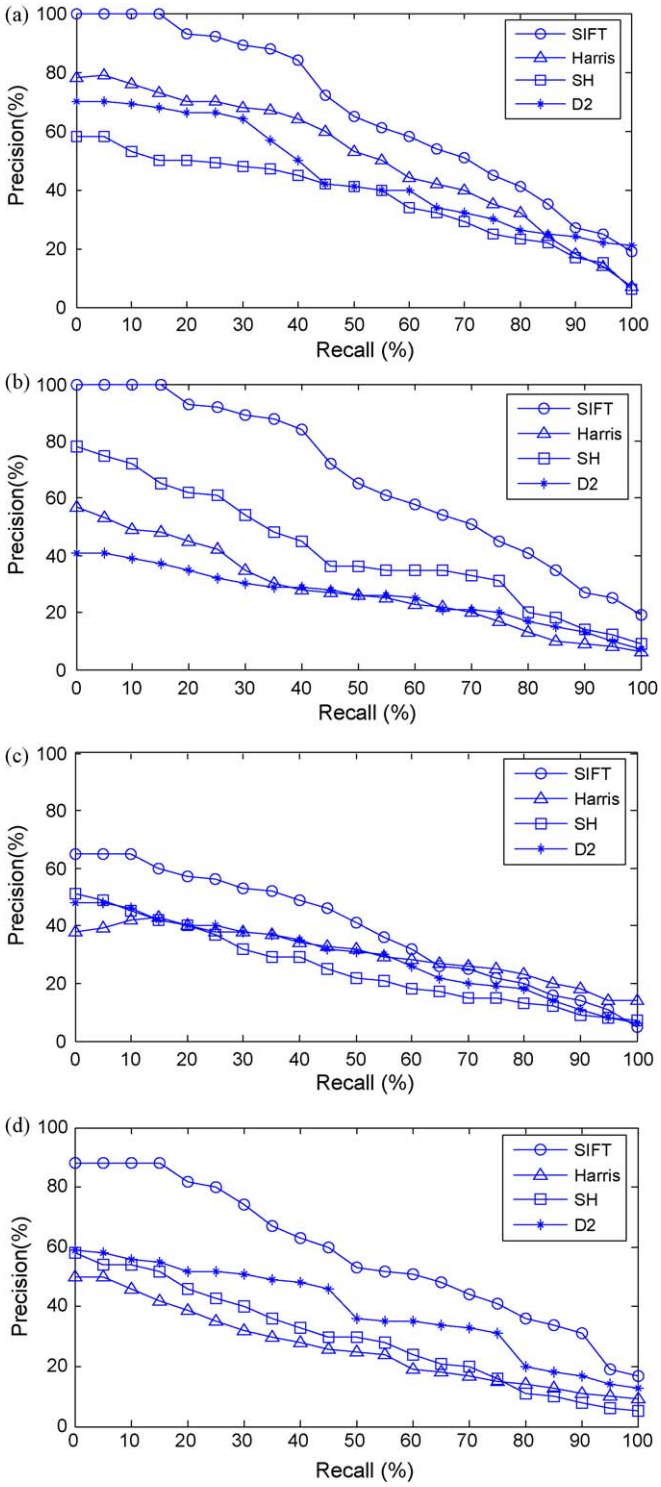

Fig. 7. The PR curve of SIFT, Harris, SH and D2 with different retrieval tests of Fig. 6. (a) PR value of query model 6(a), (b) PR value of query model 6(b), (c) PR value of query model 6(c) and (d) PR value of query model 6(d).

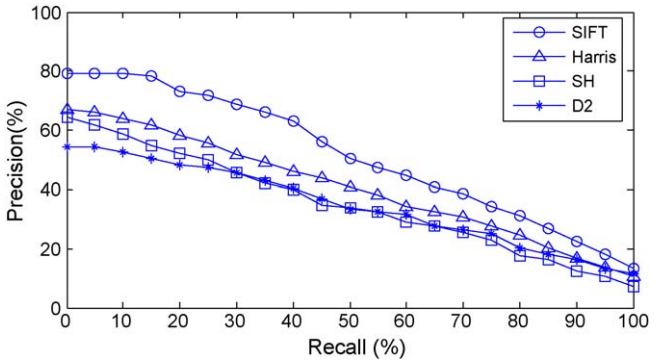

Fig. 8. Average precision/recall diagrams of all classified collection of the test database [36], obtained by using the three different feature vectors.

A summary of results obtained is presented in Fig. 8. The average retrieval performance of the proposed SIFT approach is compared to the Harris and SH and D2 approaches. The results show that the SIFT approach is significantly better than the other approaches. Harris–Laplace detector shows high repeatability and localization accuracy [39]. However its scale estimation is less accurate than SIFT due to the multiscale nature of corners. This is due to the fact that edges of 2D CAD image are better localized in scale than corners. The SH and D2 approaches are based on the 3D model analysis while the approaches are a little coarse on the feature extraction after mapping the model to the sphere. The important features, such as edges and corners, are diminished by taking the global analysis on the sphere.

4. Conclusions and future work

In this paper, a combined 3D CAD model dimension reduction and matching is presented. It maps the 3D CAD models to 2D principal images and match the 3D models to database with their 2D SIFT features. The feature matching is implemented in the 2D domain where most of the 3D structures are retained, and localized important features of CAD models such as edges and corners are extracted by the SIFT approach. This approach is shown to perform much better than the former approaches, such as shape distribution, spherical harmonics, and Harris–Laplacian affine invariant detector. Further works may be conducted to match partial shapes as the key point matching concerns local descriptors and hence, only parts of the points are needed for matching. The combinations of multiple features both from 3D and 2D may also improve the retrieval performance as could be expected.

Acknowledgement

This project is supported by the AcRF Tier 1 (R-256-000-246-112) of the Ministry of Education, Singapore.

References

[1] A. Elinson, D.S. Nau, W.C. Regli, Feature based similarity assessment of solid models, in: C. Hoffman, W. Bronsvoort (Eds.), Fourth Symposium on Solid Modeling and Applications, ACM Press, New York, NY, USA, 1997, pp. 297–310.
[2] J.H. Han, W.C. Regli, M.J. Pratt, Algorithms for feature recognition from solid models: a status report, IEEE Transactions on Robotics and Automation 16 (6) (2000) 782–796.
[3] J. Shah, D. Anderson, Y.S. Kim, S. Joshi, A discourse on geometric feature recognition from CAD models, ASME Journal of Computing and Information Science in Engineering 1 (1) (2001) 41–51.
[4] V. Cicirello, W.C. Regli, An approach to a feature-based comparison of solid models of machined parts, Artificial Intelligence for Engineering Design, Analysis, and Manufacturing 16 (5) (2002) 385–399.
[5] D. Mcwherter, M. Peabody, A. Shokoufandeh, W. Regli, Solid model databases: techniques and empirical results, ASME Journal of Computing and Information Science in Engineering 1 (4) (2001) 300–310.
[6] A. Cardone, S.K. Gupta, A. Deshmukh, M. Karnik, Machining feature-based similarity assessment algorithms for prismatic machined parts, Computer-Aided Design 38 (9) (2006) 954–972.
[7] A. Cardone, S.K. Gupta, M. Karnik, A survey of shape similarity assessment algorithms for product design and manufacturing applications, ASME Journal of Computing and Information Science in Engineering 3 (2) (2003) 109–118.
[8] M.M. Marefat, J. Qiang, Hierarchical Bayesian methods for recognition and extraction of 3-D shape features from CAD solid models, IEEE Transactions on Syetem, man, and Cybernetics. Part A. Systems and Humans 27 (6) (1997) 705–727.
[9] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, ACM Transactions on Graphics 21 (4) (2002) 807–832.
[10] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: Proc. the Eurographics, ACM SIGGRAPH Symposium on Geometry Processing (SGP 2003), Aachen, Germany, June, (2003), pp. 156–164.
[11] D.Y. Chen, M. Ouhyoung, X.P. Tian, Y.T. Shen, M. Ouhyoung, On visual similarity based 3D model retrieval, in: Proc. Eurographics, Granada, Spain, 2003.
[12] V. Jain, H. Zhang, A spectral approach to shape-based retrieval of articulated 3D models, Computer-Aided Design 39 (5) (2007) 398–407.
[13] B. Bustos, D.A. Keim, D. Saupe, T. Schreck, D.V. Vranic, Feature-based similarity search in 3D object databases, ACM Computing Surveys 37 (4) (2005) 345–387.
[14] A.Y. Ip, D. Lapadat, L. Sieger, W.C. Regli, Using shape distributions to compare solid models, in: Seventh ACM Symposium on Solid Modeling and Applications, June, (2002), pp. 273–280.
[15] D. Bespalov, A. Shokoufandeh, W. Sun, Scale-space representation and classification of 3D models, Journal of Computing and Information Science in Engineering 3 (4) (2003) 315–324.
[16] C.Y. Ip, W.C. Regli, A 3D object classifier for discriminating manufacturing processes, Computers and Graphics 30 (6) (2006) 903–916.
[17] S. Hou, K. Lou, K. Ramani, SVM-based semantic clustering and retrieval of a 3D model database, Computer-Aided Design and Application 2 (2005) 155–164.
[18] S. Hou, K. Ramani, Classifier combination for sketch-based 3D part retrieval, Computers and Graphics 31 (4) (2007) 598–609.
[19] J. Pu, K. Ramani, An integrated 2D and 3D shape-based search framework and applications, Computer-Aided Design and Applications 4 (6) (2007) 817–826.
[20] N. Iyer, Y. Kalyanaraman, K. Lou, S. Jayanti, K. Ramani, A reconfigurable, intelligent 3D engineering shape search system. Part I. Shape representation, in: ASME DETC 2003 Computers and Information in Engineering Conference, September, 2003.
[21] S. Jayanti, Y. Kalyanaraman, N. Iyer, K. Ramani, Developing an engineering shape benchmark for CAD models, Computer-Aided Design 38 (2006) 939–953.
[22] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three dimensional shape searching: state-of-the-art review and future trends, Computer-Aided Design 37 (5) (2005) 509–530.
[23] J.B. Tenenbaum, Mapping a manifold of perceptual observations, Advances in Neural Information Processing Systems, vol. 10, The MIT Press, Cambridge, MA, USA, 1998, 682–688.
[24] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[25] B. Nadler, S. Lafon, R.R. Coifman, I.G. Kevrekidis, Diffusion maps, spectral clustering and the reaction coordinates of dynamical systems, Applied and Computational Harmonic Analysis: Special Issue on Diffusion Maps and Wavelets 21 (2006) 113–127.
[26] B. Scholkopf, A.J. Smola, K.R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319.
[27] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.
[28] L.J.P. van der Maaten, E.O. Postma, H.J. van den Herik, Dimensionality reduction: a comparative review, Technical Report, MICC, Maastricht University, The Netherlands, 2007.
[29] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd ed., John Wiley & Sons, 2000.
[30] D.G. Lowe, Distinctive image features from scale-invariant key points, International Journal of Computer Vision 60 (2) (2004) 91–110.
[31] D.G. Lowe, Object recognition from local scale-invariant features, in: International Conference on Computer Vision, Corfu, Greece, September, (1999), pp. 1150–1157.
[32] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis & Machine Intelligence 27 (10) (2005) 1615–1630.
[33] G. Taubin, A signal processing approach to fair surface design, in: Proc. of ACM SIGGRAPH, 1995, 351–358.
[34] C. Schmid, R. Mohr, Local gray value invariants for image retrieval, IEEE Transactions on Pattern Analysis & Machine Intelligence 19 (5) (1997) 530–535.
[35] S. Mallat, S. Zhong, Characterization of signals from multiscale edges, IEEE Transaction on Pattern Analysis and Machine Intelligence 14 (7) (1992) 710–732.
[36] http://shapelab.ecn.purdue.edu/Benchmark.aspx.
[37] C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.
[38] K. Mikolajczykm, C. Schmid, Scale and affine invariant interest point detectors, International Journal on Computer Vision 60 (1) (2004) 63–86.
[39] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point detectors, International Journal of Computer Vision 37 (June (2)) (2000) 151–172.

Kunpeng Zhu received his Ph.D. degree in 2007 from the Department of Mechanical Engineering, National University of Singapore (NUS). He is currently a research fellow in the Lab of Concurrent Engineering and Logistics (LCEL), NUS. His research interests are in the areas of engineering design, precision machining modeling and condition morning. In these areas, he has published in refereed journals and contributed to books.

Yoke San Wong received his Ph.D. degree from the University of Manchester Institute of Science and Technology, Manchester, UK. He is currently a professor in the Department of Mechanical Engineering, National University of Singapore, Singapore. His research interests are in the areas of product design and manufacturing. In these areas, he has published in refereed journals and proceedings of international conferences, contributed to books, and participated in several local and overseas funded projects.

Wen Feng Lu is currently the associate professor of Department of Mechanical Engineering at the National University of Singapore (NUS). He received his Ph.D. in Mechanical Engineering from the University of Minnesota, USA and had been a faculty at the University of Missouri, USA for 10 years after receiving his Ph.D. degree. He later worked as the group manager and senior scientist in the Singapore Institute of Manufacturing Technology, Singapore for 6 years before joining NUS. His research interests include design methodologies, IT in product design, product lifecycle.

Han Tong Loh is an associate professor in the Department of Mechanical Engineering and also the vice dean of the Faculty of Engineering at the National University of Singapore. He obtained his Bachelor of Engineering from the University of Adelaide, a Master of Engineering from NUS and a Master of Science and a Doctor of Philosophy from the University of Michigan (Ann Arbor). His research interests are in the areas of data mining, rapid prototyping, robust design and computer aided design and he has published widely in these fields.