SIMILARITY ASSESSMENT AND RETRIEVAL OF CAD MODELS

LI MIN

NATIONAL UNIVERSITY OF SINGAPORE

2011

SIMILARITY ASSESSMENT AND RETRIEVAL OF CAD MODELS

BY

LI MIN
(B.Eng., M.Eng.)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2011

Acknowledgements

First of all, I would like to thank my supervisors, Professor Jerry Fuh Ying Hsi, and Associate Professor Zhang Yunfeng, not only for their continual supervision throughout the research of mine, but also for their kindly encouragement and helpful suggestions given to me, during the difficult times of my PhD study. Their solid knowledge, wise insight, timely feedback and careful revision ensured my research can be completed. Hereby, I would like to show the most sincere gratitude to them.

I would like to thank my Thesis Committee members for their comments and suggestions. I would also like to thank Professor Wong Yoke San and Associate Professor Lu Wen Feng for their valuable comments and suggestions during my PhD qualification examination.

In addition, I would like thank Dr. Qiu Zhiming, Dr. Feng Wei, Dr. Tan Yaxin, Dr. Huang Xingang, Dr. Lu Cong, Dr. Fan Liqing, and Dr. Zhu Kunpeng from LCEL for their generous assistance during my research. My thanks also go to Dr. Gao Zhan, Dr. Liu Zhuo, Wu Yifeng, Chen Xiaolong, Zhu Huabing, Li Haiyan, Wang Yifa, Xue Ligong, Wang Jinling, Zheng Fei, Geng Lin, Wang Yan and Zhong Xin. I really enjoyed the camaraderie of team participation and friendly atmosphere they made.

I would like to express my special gratitude to my family members, especially parents, parents in law and my wife, for their selfless and endless encouragement, understanding and love accompanying with me throughout my life.

Table of Contents

Summary

With rapid globalization and highly competitive markets, mechanical design reuse has been recognized as an effective way for manufacturing enterprises to survive by revitalizing existing designs instead of creating new ones. However, existing 3D content-based retrieval algorithms and systems, which have only focused on geometrical representations (*i.e.*, meshed or surface models), can hardly retrieve reusable results for reuse. An effective similarity assessment and retrieval mechanism for CAD model reuse, which also takes the mechanical reusability into account, has not been defined. Therefore, this research aims to develop a reuse-oriented retrieval mechanism to locate reusable CAD models effectively.

A semantics-based feature directed acyclic graph (FDAG) representation has been developed to capture complicated modeling interdependency knowledge among feature constitutes of a CAD model. Based on modeling expertise captured by FDAG representation, complicated and implicit design precedence semantics are organized as a partially ordered set (POSET). Two knowledge-driven FDAG partitioning schemes have been proposed to extract reusable CAD components. With these partitionings applied on existing CAD models, the CAD model similarity is no longer assessed on rigid 3D shapes. Instead, details of models are progressively simplified by using the proposed horizontal FDAG partitioning; therefore, assessment on essential similarity becomes possible. On the other hand, reusable sub-parts are extracted from complete models by using the vertical FDAG partitioning.

An essential shape matching (ESM) method supporting CAD model retrieval based on their essential shape similarities has been presented. In ESM, complete CAD models are simplified, and their essential shapes are preserved for comparison. An essential shape aggregation (ESA) descriptor has been defined for comparing only essential shapes of CAD models while effectively tolerating trivial details.

A partial shape matching (PSM) method has also been proposed to address the reuse-oriented retrieval of CAD partial components. In the PSM method, the vertical partitioning has been applied to find out disjointed sub-graph from the FDAG representation, by examining the reachability of a POSET data. The found disjointed sub-graphs are equivalent to reusable CAD partial components, which are further compared by the partial shape aggregation (PSA) descriptor.

A prototype system has been implemented to demonstrate the feasibility of the proposed reuse-oriented retrieval method. The effectiveness has also been evaluated on more than six hundred realistic CAD models and multiple case studies. The proposed method brings more advantages: (1) it offers ease of reuse on retrieved results as the reusability is taken into account in the retrieval; thus, inflexibility to reuse can be greatly avoided, and (2) it maximally preserves design intelligence to reused parts. The prototype provides users the access to original modeling expertise embedded in existing models when reusing. As a result, design intelligence including parametric constraints will be inherently transferred to new designs and future reuse.

List of Figures

List of Tables

Nomenclature

2D	Two dimensional
3D	Three dimensional
AABB	Axis-aligned bounding box
ANN	Artificial neural network
AP	Articulation point
API	Application programming interface
BFS	Breadth-first search
B-Rep	Boundary representation
CAD	Computer-aided design
CBIR	Content-based image retrieval
D2	Function indicating distance between two random points on object surface
DAG	Directed acyclic graph
DBMS	Dilation based multi-resolutional skeleton
$deg+(v)$	The number of direct successors of a graph node (*i.e.*, out-degree)
$deg-(v)$	The number of direct predecessors of a graph node (*i.e.*, in-degree)

DFS	Depth-first search
DOT	Dot language
ESA	Essential shape aggregation descriptor
ESM	Essential shape matching
FDAG	Feature directed acyclic graph
GT	Group technology
GUI	Graphical user interface
LOD	Level of detail
L_1^{max}	Theoretical maximum Manhattan distance between two histograms
MRG	Multi-resolutional Reeb graph
NP	Non-deterministic polynomial time
NURBS	Non uniform rational basis spline
OEM	Original equipment manufacturer
PDM	Product data management
PFM	Parametric and feature-based modeling
PNG	Portable network graphics
POSET	Partially ordered set
P-R	Precision-recall curve

PSA	Partial shape aggregation descriptor
PSM	Partial shape matching
P^{λ}	Precision value at a certain recall λ rate
SD	Shape distribution
SH	Shape histogram
STEP	Standard for the exchange of product model data
SVD	Singular value decomposition
TC	Transitive closure
UDF	User-defined feature
UML	Unified modeling language
WCS	World coordinate system
XML	Extensible markup language

Chapter 1 Introduction

1.1 Background

Today's manufacturing industry becomes highly competitive due to the increasing globalization of the economy and the rapidly changing demands from customers. Reducing costs and shortening time to market are key initiatives to survive for an enterprise in such intensive market competition. Reusing existing designs instead of building them from scratch will play a key role in helping enterprises succeed in the market competition. An effective design reuse, which maximally revitalizes existing designs, will significantly benefit manufacturing enterprises in the following aspects:

- **Reduction of development costs.** Considering the cost of creating a new mechanical part from scratch, 48% of total expenses will be spent on design. This number means that every successful reuse can cut development cost by half [PSMC 2002]. Therefore, design reuse increases the return on investment (ROI) significantly.
- **Saving of product lead-time.** A significant percentage (80%) of previous mechanical designs can be reused to facilitate new designs, either by selecting one existing part that directly meets new requirements or making minor modifications to them [Gunn 1982].

Although many existing parts are theoretically available to be reused for new designs, accurately locating of a desired part from a large archival repository is not straightforward [Jackson and Buxton 2007]. In an enterprise-level design repository, there are hundreds of thousands of archived parts. It will be an extremely compelling task for designers to find out the desired one among these innumerable parts, without a convenient retrieval tool.

1.1.1 Manual classification and retrieval

In early years, mechanical engineers had attempted to organize archived mechanical designs manually for future retrieval. Group technology (GT) [Mitrofanov 1966] is probably the first effort to analyze and manage 2D draft works in a systematic way. The philosophy of GT is that various mechanical parts having similarities are grouped together to achieve a higher level of commonality integration. Various mechanical attributes can be considered as GT similarity, such as design properties [Iyer and Nagi 1994], manufacturing properties [Lee and Fischer 1999] and process planning properties [Herrmann and Singh 1997]. These attributes are encoded into a sequence of alphanumeric strings (*i.e.*, GT codes). All parts are hierarchically classified into families according to the GT code similarity. If a user wants to retrieve a part with specific shape properties, he or she can generate the GT code of the desired part, and look into part families which have similar GT code because parts within a family have higher reuse significance. Moreover, retrieved parts within a family normally share similar manufacturing processes, and they can be manufactured in the same machine cell, thus facilitating cellular manufacturing.

Nevertheless, manual classification and retrieval works face the following problems: the manual process is slow, expensive, and error-prone. The GT approach also has the same issues. Firstly, building a GT database might not be automatic or programmable due to the fact that GT code generation is still repetitive work which heavily relies on eye-labeling. Even for experienced engineers, their speed of the manual coding rarely exceeds hundred parts per day. Secondly, plenty of manpower will be occupied to maintain a large-sized GT repository's consistency and accuracy. Thirdly, manual coding involves individual interpretations, which are prone to errors. Last but not least, GT methods only work well for relatively simple parts, *e.g.*, sheet metal or rotational ones. With these limitations, manual retrieval systems like GT approaches are unable to handle heavy amount of parts well as maintenance and running costs will be too high to be affordable if manual works are involved [Love and Barton 2001].

1.1.2 Metadata based tagging and retrieval

In recent years, with the emergence of affordable personal computers, new computer-aided design (CAD) techniques and tools have been provided to help designers streamline mechanical design. Especially with the introduction of parametric and feature-based modeling (PFM) [Shah and Mäntylä 1995] in early 1990s, the PFM modeler allows users to intuitively build models by using semantic features, and flexibly adjusting geometric parameters of them.

PFM has significantly changed product design paradigm and enabled rapid creation and modification of variant instances [Mäntylä *et al.* 1996]. As a consequence, nowadays the magnitude of mechanical part variations is increasing at a staggering speed, and product data management (PDM) systems have been employed to organize such heavy amount of data [Miller 1998]. In most of commercial PDM systems [SmarTeam 2006, Windchill 2006], part management functions are realized by a metadata based tagging and retrieval mechanism. Metadata is a concept traditionally used in library catalogues to describe contents of books. In modern PDM context, metadata is a term to cover textual attributes that are assigned to product-related documentation. A specific kind of metadata can be considered as a tag. With certain tagging during part creation or modification, prior designs can be organized as a multi-categorical model in PDM systems. Established categories with tags enable users to search for a needed design with exact or inexact keyword matching, such as filenames, profile classifications, materials, or other customized keywords.

However, the accuracy of such metadata-based retrieval methods is subject to consistent perceptions of annotators. Although metadata-based retrieval works well for standardized parts, inconsistent naming issue will become significant, especially under complicated engineering contexts, in which the annotation process is prone to be ambiguous and incomplete [Min *et al.* 2004]. As a result, a recent industrial survey has pointed out that, 46% of manufacturing companies, even top performers on design reuse, had complained that "users cannot find models to reuse" is still a major challenge to them [Jackson and Buxton 2007].

Since conventional CAD model retrieval methods, either manual or metadata-based, are laborious and less precise, the content-based automatic similarity assessment and retrieval technique has been proposed as an alternative solution, to retrieve CAD models in a more precise way [Gupta *et al.* 2006].

1.2 Automatic Content-Based Similarity Assessment and Retrieval

Positioned as an application of 3D graphics to the information retrieval (IR) problem, the automatic content-based 3D similarity assessment technique [Cardone *et al.* 2003, Bespalov *et al.* 2005, Iyer *et al.* 2005b] aims at retrieving 3D shapes by their actual contents instead of metadata-based annotations, which will not suffer from the inconsistent naming issue.

In content-based similarity assessment, there are two different kinds of 3D object's similarities [Veltkamp and Latecki 2006]: generic and partial shape similarities. The former assesses how visually similar 3D objects are, while the latter tries to find a shape of which a part that is similar to portions of another object. This division is also applicable to the similarity assessment of CAD models, and each similarity definition has corresponding applications in CAD model retrieval. The generic similarity assessment helps to retrieve general CAD models; while the partial shape similarity concentrates on the retrieval of partial CAD components.

The following sections will investigate the applicability of previously reported automatic content-based methods on CAD model retrieval applications, and evaluate how effective they are for reuse-oriented retrieval. These sections will not serve as a complete review on the content-based retrieval, as an extended literature review will be given in Chapter 2, where different types of automatic content-based retrieval methods will be reviewed and compared.

1.2.1 Retrieval of general CAD models

With the great convenience introduced by the PFM technique, designers are enabled to create a part family consisting considerable varieties to satisfy different functionality requirements. These varieties can be generated through design parameterization, which normally share a common overall shape with differential minor geometric variants. If designers are asked to create a new part with the same basic shape but having a slightly different variant, presenting existing varieties with similar shape will help them choose one of them to reuse. Therefore a common application of CAD model retrieval is to match parts based on their essential shapes because in the above scenario, the essential shape is a critical criterion in searching parts for design reuse.

However, most of current 3D retrieval methods are rigid shape based, which only compare CAD models as a whole. They lack the ability to suppress insignificant details in the assessment, and therefore they cannot tolerate such minor variants to evaluate the overall shape similarity [Hou and Ramani 2008]. This would be a major gap to retrieve CAD models based on their essential shapes, eventually preventing part family retrieval and redesign.

Furthermore, most of current 3D retrieval methods work on geometric representations, such as meshed or surface models. It means that retrieved results from those methods are surface models, which are not easy to be manipulated for reuse. This is also confirmed in an industrial survey [Jackson and Buxton 2007]. The survey reports that even for results retrieved by content-based retrieval tools, the inflexibility to reuse and the failure after modifications are major obstacles to perform reuse successfully.

To address the gap between retrieval and reuse, an ideal CAD model retrieval tool should take mechanical reusability into account, on top of other considerations, such as geometric similarity.

1.2.2 Retrieval of partial CAD components

Besides the above reuse scenario supporting general CAD model retrieval, there is another scenario that designers may want to retrieve all existing designs that share a particular CAD component. An example is illustrated in Figure 1-1, where five models share a common tapered head, which is a partial CAD component. However, complete models are dissimilar as they have considerably different locating bases. These partial CAD components normally are standardized sub-parts within an organization and therefore having high reusability for future design. In order to retrieve portions of CAD models, a partial matching method should be developed, to match only similar portions as opposed to a full object [Tangelder and Veltkamp 2008].

Figure 1-1. Different locating pins sharing a similar tapered head

In the partial matching problem, 3D segmentation plays a key role which splits the complete 3D objects into a set of pre-determined segments. Some researchers have summarized 3D segmentation techniques for CAD applications [Agathos *et al.* 2007]. In the boundary mesh segmentation study [Suzuki 2004], the author detected a boundary between segments if there is a sharp change of curvature. Furthermore, partial shapes, in the form of segmented surfaces, were matched in a many-to-many manner [Bespalov *et al.* 2006]. Another group of methods apply the clustering technique on 3D volumes in which 3D objects are either segmented in a paralleled way [Biasotti *et al.* 2006] or a symmetric way [Bespalov *et al.* 2003b]. However, all these previous works only adopted purely geometric attributes as the segmentation criteria, which are subject to minor changes of the shapes being segmented. More importantly, these methods may produce segmentations that are meaningless to CAD model reuse, *e.g.*, surface patches or shape fragments. Such results are "dump" surfaces or solids, which are of little value in reuse as the direct reuse of freeform surfaces is still challenging [Zhao *et al.* 2009]. Therefore purely geometric criteria that are used in current segmentation algorithms appear to be an obstacle to realize effective partial shape retrieval. An ideal partial shape retrieval algorithm should be able to extract reusable sub-parts from complex CAD models automatically, and effectively support direct reuse of extracted CAD sub-parts.

1.3 Research Objectives

The investigation in section 1.2 shows that current similarity assessment algorithms on 3D objects cannot support effective retrieval for design reuse. Several gaps are identified for two common retrieval scenarios: retrieval on general CAD models and partial CAD components. The objectives of this research are to address the identified obstacles, and to develop effective approaches to support reuse-oriented retrieval of CAD models. The research will be focusing on the following areas:

(1) To elaborate a semantics-based representation for 3D CAD models. In order to have a semantic representation that can effectively support future CAD model similarity assessment and retrieval, the following need to be developed:

- To identify related modeling knowledge that is most important to CAD model similarity assessment and retrieval.
- To develop an automatic acquisition mechanism to extract the identified modeling knowledge, from multifaceted mechanical information of archived CAD models.
- To present a suitable representation to capture the extracted modeling knowledge. An organized and structural representation needs to be developed to provide better views on complicated modeling knowledge.

(2) To develop an approach to support CAD model retrieval based on essential shape similarity. In order to realize the essential similarity based CAD model retrieval, the following need to be developed:

- In order to serve the assessment on essential shapes instead of complete models, significant modeling constitutions will be determined by using order theory's definition of maximal and minimal elements on a partially ordered dataset.

- To address the essential shape similarity assessment, a horizontal partitioning mechanism on the knowledge-based representation needs to be proposed to decompose the graph from minimal elements to maximal ones.

- Corresponding to the horizontal partitioning, a multi-level simplification algorithm needs to be developed to simplify complex CAD models progressively while maintain the essential shapes of the models.

- To define the essential shape similarity in CAD modeling context. Based on the similarity, an essential shape matching algorithm should be elaborated to perform essential similarity assessment and retrieval of CAD models.

(3) Development of a method to support CAD model retrieval based on partial shape similarity. In order to realize mechanical sub-part retrieval, the following need to be developed:

- In order to serve the assessment on partial shapes, a vertical partitioning scheme needs to be presented to find out disjointed portions from the modeling knowledge representation by studying the reachability on the representation.

- Corresponding to the vertical partitioning, a CAD sub-part decomposition algorithm needs be developed to segment reusable components from a complete mechanical design.
- To define the partial shape similarity. Based on the similarity defined, a partial shape matching algorithm should be put forward to address similarity assessment and retrieval on partial CAD components.

(4) Implementation of a reuse-oriented retrieval system framework to bridge the gap between retrieval and reuse. In order to realize a prototype system to prove the reuse-oriented retrieval, the following need to developed:

- To provide a convenient interface for query composition. The interface should contain an intuitive method to compose 3D searching query in a user-accustomed way.
- In essential and partial shape matching stage, the matched CAD models should be proactively promoted to designers, to shorten the processing time from querying to retrieving.
- In reuse stage, redesign suggestions should be automatically generated to help designers to consider most feasible re-design modifications.

1.4 Organization of Thesis

The aim of this chapter is to present the general background to the research work that will be described in the following chapters. It examines several common reuse cases. By analyzing the common reuse cases and existing 3D retrieval methods, it also clearly identifies some gaps between what the design reuse wants and what existing retrieval methods can offer. Research objectives are determined to bridge the found gaps between the CAD model retrieval and reuse.

Chapter 2 gives a comprehensive investigation of the previous works on 3D model retrieval. Chapter 3 presents the theoretical framework of this research, which consists of a knowledge-based representation of CAD models and its acquisition and construction approaches. In Chapters 4 and 5, two reuse-oriented CAD model retrieval methods are proposed to search for reusable CAD models based on their general and partial shape similarities, respectively. Chapter 6 describes experimental results using the proposed algorithms, and discussions. The last chapter presents the conclusions and recommendations for future work.

Chapter 2 Literature Reviews

This chapter reviews previous works on 3D matching methods. According to the shape similarity classified by Veltkamp and Latecki [2006], there are two remarkably different kinds of 3D similarity which are widely studied: generic and partial shape similarity. The former assesses how visually similar 3D objects are, while the latter tries to find a shape of which a part that is similar to a part of another object. Section 2.1 reviews previous works on 3D matching approaches based on generic shape similarity and section 2.2 reviews those on partial shape similarity.

2.1 Generic Similarity Based 3D Model Retrieval

CAD model retrieval is an important application of information retrieval. Traditional manual CAD model retrieval heavily depends on human perceptions of the mechanical part similarity. One of manual approaches is group technology (GT), which is known to the time-consuming and error-prone; thus, manual approaches can hardly manage hundreds of thousands of mechanical parts in an enterprise level.

In recent years, automatic content-based 3D retrieval methods have emerged to search CAD models in large-scale databases [Gupta *et al.* 2006]. Shape descriptors in these automatic algorithms play a key role in enabling the search. A shape descriptor is a concise, mathematical representation of 3D objects to enable them searchable, and each algorithm has its own descriptor to represent the *generic* *similarity* of 3D CAD models. Commonly used descriptors can be categorized into three main types: mathematics, visual, and knowledge-based.

2.1.1 Generic similarity retrieval by mathematics based descriptors

A direct way to characterize a 3D object is to capture its geometrical properties by mathematical representations. This section reviews these mathematics based descriptors.

Global Feature Approach

Global feature based descriptors have been proposed to extract feature vectors of global geometries of 3D models to characterize the shapes. Paquet *et al.* [2000] firstly proposed a search engine that characterizes 2D visual objects by bounding box descriptor, and compares 3D shapes using these global features: cords-based vector set, and wavelet transform-based volume occupancy. Some researchers [Elad *et al.* 2001, Zhang and Chen 2001b] put forwarded 3D matching methods to take moments of 3D solids as characteristic descriptors. Lou *et al.* [2004] proposed a method to adopt invariable principal moments to assess 3D similarity. Zhang and Chen [2001a] further extended global feature based descriptor by putting volume-to-surface ratio, moments invariants, and Fourier transformation coefficients into the 3D similarity computation. Sun and Qamhiyah [2003] proposed a method to use discrete wavelet transforms to represent curvatures of face regions and radial distances of faces boundaries. Moreover, Kazhdan *et al.* [2004] especially studied a descriptor that represents the reflective symmetry of a 3D model, which shows strong robustness against noises and sampling resolutions. A common pitfall of global feature based descriptors is that a type of feature vector can address one aspect of geometrical properties (*e.g.*, moment invariants, surface area to volume ratio etc.); therefore, single feature vector based descriptor cannot capture shape contents comprehensively. As a valuable application in data and signal processing and recognition of 3D shapes, spherical harmonics functions which approximate 3D shapes with finer harmonic coefficients, thereby it might capture more shape contents. Many studies adopted spherical harmonics based methods to evaluate the geometric similarity between 3D shapes [Saupe and Vranić 2001, Vranić *et al.* 2001]. Vranic *et al.* [2001] proposed a 3D shape descriptor based on spherical harmonics functions, where the functions are used to represent global feature vectors. In the method, feature vectors are extracted from normalized models using spherical Fourier coefficients, and compared to the spherical harmonic feature vectors of the query model. Similarly, Saupe and Vranić [2001] adopted both moments and spherical harmonic functions to assess the geometrical similarity of 3D models. Recently, the spherical harmonics based algorithm [Morris *et al.* 2005] has been extended to biological macromolecules domain to determine protein structures which have no available biochemical characterization. The algorithm adopts spherical harmonics coefficients to characterize the shape of a protein's binding pocket, and the binding pocket shape similarity is assessed as the geometrical distance in coefficient space. In more recent years, Papadakis *et al.* [2007] presented a 3D shape descriptor which adopted spherical harmonic functions to compute scaling and axial flipping invariance of the descriptor, and continuous principal component analysis (PCA) is used to achieve the rotation invariance of the analyzed models.

Statistical Approaches

Another group of mathematics based descriptors has been proposed to describe 3D geometries using statistical algorithms. The idea of statistics-based methods is "count and accumulate", that is, statistics measures of 3D objects are sampled and counted into a fixed length histogram in an accumulative manner. The histogram is then used to assess 3D similarity. The method based on shape distributions was firstly reported in [Osada *et al.* 2001]. A shape distribution (SD) descriptor represents a probability distribution of the overall shape of a 3D model. The probability distribution is randomly sampled by a geometric function, *e.g.*, A3 measures the angle among three random points on the model surface, and D2 stands for the distance between two random points on the surface. By comparing probability distributions between two models, similarity assessment is achieved. In order to increase discriminating capability for detailed parts, Ip *et al.* [2002] extended the original shape distribution algorithm by subdividing the D2 function into three types: IN, OUT, and MIXED. More recently, enhanced shape functions have been proposed in [Ohbuchi *et al.* 2005, Hou *et al.* 2007]. Another kind of 3D shape statistical function is the shape histogram (SH), which evolves from the section coding technique used to retrieve 2D polygons. The basic idea of SH is to partition the 3D space and encode on partitioned models. Three basic partition techniques were introduced by Ankerst *et al.* [1999], namely shell partitioning, pie partitioning, and spider-web partitioning. The shell partitioning splits the space with concentric sphere with variable radius; while the pie-like segmentation projects planes passing the centre of a sphere. The spider-web one is a combination of the former two. Spatial percentages occupied by the model in each partition are encoded into a vector, which is used for similarity assessment. A variable SH method was reported in [Kriegel *et al.* 2003], which uses a voxelized representation instead of the original model. Because computing of whether a voxel resides in a partition or not is straightforward, the computational cost is reduced. Other statistics-based descriptors include a cord-based measure [Paquet and Rioux 1999], scalar function distributions [Gal *et al.* 2007], sphere projection signature [Leifman *et al.* 2005], and density-based descriptor [Akgul *et al.* 2007].

The mathematical descriptors investigated above, both statistical and global feature ones, are computationally efficient in comparison because the mathematical characteristics are represented by fixed-length vectors or histograms. In addition, statistical SD descriptor shows desirable invariance to rotation and translation, and has satisfactory discrimination for primitive shapes. Nevertheless, as characteristics are sampled discretely, 3D details of shapes might not be always sampled, which results in a fact that these descriptors cannot discriminate details effectively. Most importantly, the proposed mathematical characteristics do not include human perception on visual similarity. In other words, these pure mathematics based shape descriptors are only computer-understandable; however, they lack a straightforward explanation in human perception.

2.1.2 Generic similarity retrieval by visual based descriptors

Compared with the shape descriptors reported in section 2.1.1, which are completely captured by mathematical characteristics, human beings may use a different perception to define 3D shape similarity. In human perception, two 3D objects are similar if they are looked alike from every side, or one bears a strong resemblance to another in terms of 3D visual structure.

2D View Approach

In the past decades, content-based image retrieval (CBIR) techniques have been extensively studied. CBIR algorithms have been developed to search for similar 2D images, in which the term content may refer to colors, textures or profiles that are derived from images [Veltkamp and Tanase 2002]. However, a number of CBIR algorithms have been adopted and extended to compare 3D models in the shape retrieval research. These image-based retrieval algorithms search similar shapes by comparing characteristic 2D views of 3D objects. The basic idea of image-based shape retrieval is that two 3D objects are similar if they look alike from all viewing angles. Various 2D views have been chosen to characterize shape models, and characteristic images are then used for similarity assessments. Some researchers [Funkhouser *et al.* 2003, Wang and Cui 2004, García *et al.* 2007] have adopted silhouettes as characteristic images. Chen *et al.* [2003] proposed an enhanced silhouette descriptor that is characterized from multiple projected silhouette views with light-fields, which are uniformly distributed on a bounding sphere. The light-field descriptor might effectively filter out high-frequency noise. Furthermore, orthogonal images of 3D models, where darker pixels indicate longer distances from viewing planes to the objects, are utilized to compare the shape similarity by Vranic [2004]. In 2005, Barton *et al.* [2005] adopted orthogonally projected profiles to enable mechanical model retrieval. The impacts of freehand sketch variability on the retrieval performance were also discussed. Pu *et al.* [2006] also represented mechanical drawings as a level of detail (LOD) based descriptor which consists of 2D characteristic on three distinct levels of details – silhouette, contour, and drawing. A combination of different LOD sectional images has then been adopted to compute the similarity between CAD models. Furthermore, Hou and Ramani [2008] introduced a divide-and-conquer method to match feature point correspondences of 2D contours in order to compare deformable models. The above image-based methods have naturally enabled designers to submit 2D views as search queries, which were reported by Funkhouser *et al.* [2003] and Love *et al.* [2004]. Pu *et al.* [2006, 2007] also adopted the freehand sketching as one of 3D query methods, which is quite straightforward.

3D Graph Approach

A graph is a natural choice for capturing 3D visual topology, and graph-based descriptors have been adopted to characterize structures of CAD models. In general, graph-based descriptors can be grouped into three types: boundary representation (B-Rep), Reeb, and skeletal based. B-Rep graph based approaches compare CAD model similarity based on their boundary representations. B-Rep descriptors are represented by undirected graphs, and the similarity comparison is converted to the graph matching problem. El-Mehalawi and Miller [2003b, a] reported their work that used

B-Rep entities of a STandard for the Exchange of Product (STEP) model to construct an attributed graph. In their graph-based representation, graph nodes are converted from adjacent B-rep faces of a CAD model, and edges between the faces are links between the nodes. However, direct B-Rep graph comparison is impractical as a basic mechanical part may have a too complex B-Rep graph to compare within polynomial time as graph matching is a classical NP-complete problem. Instead, eigenvalues of B-Rep graphs are compared in [McWherter *et al.* 2001]. Another group of methods is to capture 3D models using Reeb graphs. Hilaga *et al.* [2001] proposed the first method to capture 3D topology by a multi-resolutional Reeb graph (MRG). The basic idea of Reeb graph method is like follows. Firstly, a 3D model is faceted to prepare for the Reeb graph generation. Secondly, the faceted model is horizontally sliced into a number of partitions. Each sliced partition is regarded as one node in an MRG. The adjacency between partitions is mapped to the adjacent edge between Reeb graph nodes. More recently, Reeb graph methods have been extended to retrieve complex CAD models [Chen and Ouhyoung 2002, Bespalov *et al.* 2003a]. However, Reeb graph based technique is found to be sensitive to surface connectivity of the facet models. The third group of graph-based shape descriptor is using skeletal graphs. Skeletons are the simplified geometry representation of 3D shapes, which can be obtained by topologically preserved thinning algorithms [Xie *et al.* 2003, Klette and Pan 2005]. Skeleton graph based methods simplify 3D models in a topology preserving way. Skeletal graphs are built upon the simplified topologies (*i.e.*, skeletons), and skeletal graph resemblance is used to assess 3D similarity. Lou *et al.*

[2005] assessed 3D similarity by comparing eigen-values of skeletal graph adjacency matrix; while the combinational assessment of geometric and graphical measures [Iyer *et al.* 2005a, Gao *et al.* 2006] have been proposed for 3D similarity comparison. In addition, both global similarity and local similarity of 3D shapes have been considered in several studies [Shokoufandeh *et al.* 2005, Zhang *et al.* 2005, Gao *et al.* 2006]. In 2007, Ju *et al.* [2007] proposed a redundant feature pruning algorithm to generate better skeletal representation. Especially, comparison methods based on critical point correspondence have been studied using deformable 3D objects and 2D drawings [Tam and Lau 2007, Hou and Ramani 2008].

The above structural descriptors are intuitive as they captured shape topologies, which is similar to human perception of visual similarity. Nevertheless, these descriptors cannot be directly adopted into CAD model retrieval because the CAD model similarity is not equivalent from the visual resemblance. In mechanical design domain, each CAD modeled design has specific mechanical properties. Shape descriptor may not be effective if these mechanical and design aspects of knowledge are not properly considered into CAD model retrieval. Moreover, these descriptors have specific limitations. In 2D view based descriptors, high-level 3D information can be lost during the conversion from 3D shapes to 2D images. Moreover, choosing characteristic 2D views for a 3D object is not deterministic. Several view clustering techniques reported by [Cyr and Kimia 2001, Ansary *et al.* 2007] attempted to determine optimally characteristic views. As for the 3D graph based algorithms, they have trade-off between the comparison accuracy and complexity: the accuracy of comparison is highly dependent on the granularity of characterized graphs. A finer granularity graph requires more time to generate and compare, which might not be responsive; while coarser graphs cannot capture 3D topologies effectively.

2.1.3 Generic similarity retrieval by knowledge based descriptors

To address the lack of knowledge issue in purely geometrical retrieval, in CAD model retrieval domain, mechanical knowledge specific to the domain has been considered in several studies. In works of Cardone *et al.* [2004, 2006], a manufacturing feature facilitating mechanism is employed to estimate costs of machining new parts, by analyzing machining alignment and orientation information of existing prismatic parts. The basic idea of the study is that visually similar parts share similar machining processes; thus, according to the expense on a known machining process, the cost for a new part can be estimated based on how similar the new and the known are. Furthermore, the mechanical design similarity is assessed using the sub-graph isomorphism of machining feature graphs [Cicirello and Regli 2002], where access directions, types, volumes, tolerances and group cardinality of machining features were used as similarity descriptors being compared. The aforementioned methods only took account of manufacturing semantics, while not considered design features that represent design-related semantics.

In recent years, some researchers adopted recognized design features to enable CAD model retrieval. Hong *et al.* [2005] presented a multi-step method for CAD model retrieval. In the first step, detailed design features, *e.g.*, concaves, blends, passages are recognized and suppressed by wrap-out or smooth-out operations that were introduced by Kim *et al.* [2005]. With specific features suppressed, a coarse pre-classification is applied to classify models because overall appearances after suppression are more differential. In the second step, feature information, *e.g.*, delta volume and delta face number of features, are integrated into high-level similarity assessment. In their extended work [Hong *et al.* 2006], a semi-random sampling algorithm is introduced to have better pre-classification and find more similar designs. In the study of Chu and Hsu [2006], the design feature information is extracted from mechanical models. The extracted features are used to generate a colored graph where white nodes stand for additive features and black nodes stand for subtractive features. Both colored feature graphs and geometrical properties are compared to those of another model, to determine the CAD model similarity. Cheng *et al.* [2007] extended the work of Chu and Hsu by putting forward an artificial neural network (ANN) based method to provide guided feedbacks to the original knowledge-based similarity assessment algorithm and adjust on initial ranking inaccuracy. This series of research might have provided a direction for 3D search because it could effectively bridge the discrepancy between human perception and machine-learnt similarity.

However, the aforementioned algorithms only take recognized design features into similarity consideration. Feature recognition results might be ambiguous and uncertain due to the multiple-interpretation issue. Compared with recognized features, design features created by designers during modeling are more valuable as it conveys the information that how models were built. The designer-defined modeling expertise, such as modeling precedence and dependency, would have more significant impacts on instructing how to retrieve similar and reusable CAD models to serve reuse-oriented retrieval purpose.

2.2 Partial Similarity Based 3D Model Retrieval

As opposed to generic similarity, partial shape similarity assesses at the portion level to match resembled parts of a complete 3D object. In reuse-oriented retrieval paradigm, designers may only want to search for a sub-part instead of a complete design as redesign reference. Therefore, partial similarity assessment should also be taken into account for practical CAD model retrieval.

3D model segmentation plays a critical role in enabling partial shape retrieval by extracting meaningful sub-parts from 3D objects. For instance, if a user is searching the wings of a plane, segmenting wings from planes is a compulsory step. In early years, 3D segmentation techniques had been widely studied for medical volumes to address anatomical organ extraction [Lakare 2000]. Medical volume segmentation enables physicians to extract portioned views of human organs from complex scanned anatomical structures; in this way, doctors can only focus on regions of interest. In recent years, 3D segmentation has another application in the content-based retrieval. Once all sub-parts are segmented, geometrical and structural characteristics of the segmented portioned can be captured by shape descriptors and compared with partial shape queries. It means that the attempt trying to retrieve 3D objects based on partial similarity will return to a basic retrieval problem if an appropriate 3D segmentation method can be applied. In the following sections, previous works on 3D segmentation techniques and related applications in partial shape retrieval will be reviewed.

2.2.1 Partial similarity retrieval by stochastic techniques

Stochastic techniques are also applied on 3D volume segmentation. Common stochastic techniques used include *classification* and *clustering*. Classification is a supervised process similar to labeling, and tries to examine and categorize 3D segments based on known local geometry characteristics and hierarchical topology maps; while the term clustering is defined as the unsupervised process of grouping 3D data into segmentations whose members show strong spatial resemblance. The classification is extremely useful for anatomical structure extraction as anatomical structures are stable and unchanged. Some researchers proposed the classification based methods to perform automated classification and segmentation on scanned 3D brain models and label complicated cortical surfaces [Sandor and Leahy 1997, Jaume *et al.* 2002]. Shamir *et al.* [2003] proposed a classification based segmentation method on categorical models that have pre-defined topological hierarchy, such as fingers as ridges are always connecting to the hand palm that is like blob primitives. Therefore, partial correspondences are greatly facilitated by the known geometrical and topological characteristics. However, classification is not suitable for automatic segmentation of arbitrary 3D models which do not have pre-defined characteristics.

Clustering based methods come under the unsupervised class of 3D segmentation algorithms. Katz and Tal [2003] examined probabilistic clustering application on 3D model segmentation. In their research, a fuzzy k-means clustering algorithm was adopted to decompose 3D polygonal models, and the clustering probability (*i.e.*, segmentation criterion) is the combination of angular and geodesic distance between faces. In the study of Bespalov *et al.* [2003b], the singular value decomposition (SVD) clustering scheme [Thomasian *et al.* 1998] was applied on the geodesic distance of surfaces to produce the scale-space representation of 3D models. Based on the bisectionally clustered scale-space tree, similarities between models are compared in a divide-and-conquer manner: the similarity of tree nodes could be estimated based on the resemblance of their sub-trees. Bespalov *et al.* [2006] extended their previous works to address partial shape retrieval problem. In their attempt for polyhedral model segmentation, a new distance function, namely maximum angle on angular shortest path is defined, to describe local surface smoothness. Once the smoothness of local surfaces is identified, the recursive bisectional decomposition segments models into surface patches with interactive control of the decomposition termination, and partial matching can be conducted on the segmented patches. Similar studies on clustering based segmentation include [Liu and Zhang 2004, Klasing *et al.* 2008, Li 2010], which apply various 3D distance criteria and clustering techniques on 3D model data.

In the aforementioned approaches, stochastic segmentation techniques can generate superior results for certain models. For instance, classification based methods work well on categorical models that have predetermined geometrical or topological characteristics; while the methods using bisectional clustering techniques will outperform others on segmenting symmetrical models. Moreover, clustering methods need a pre-defined parameter as algorithmic input, *i.e.*, the number of clusters to be generated, which cannot be easily determined for all types of 3D models.

2.2.2 Partial similarity retrieval by structural techniques

Another popular 3D segmentation criterion of segmentation is 3D objects' structural properties. Structural segmentation techniques try to find differentiating properties of structures. One of structural criteria is 3D boundary, which is the intersection of two surface regions that might have various features, such as curvatures or intensities. Suzuki [2004] reported a boundary based segmentation method in which a boundary can be detected if there is a sharp change of surface smoothness. The smoothness is defined as angular difference between adjacent mesh triangles in the method. Suzuki *et al.* [2005, 2006] extended their works to build a partial shape retrieval system based on mesh curvature. One advantage of the boundary detection technique is that it works well on datasets with obvious contrasts between different regions, *e.g.*, linking vessels of a vessel tree. Moreover, convex hull is another structural segmentation criterion, which has important application in collision detection, and the same concept had been adopted for segmenting characters [Chang and Chen 1999] in 2D domain. Zuckerberger *et al.* [2002] proposed a convex hull flooding algorithm in 3D context, which supports non-convex segment generation. A post-processing method was described to merge small segmented patches into bigger one, which can generate smaller amount of segmentations results. By using the proposed flooding and post-processing, every 3D model can be decomposed into a small number of patches, and searching on patches instead of models is possible. The third segmentation criterion is the branch intersection of Reeb and skeleton graph structures. Reeb graph [Hilaga *et al.* 2001, Bespalov *et al.* 2003a] and skeleton graph [Iyer *et al.* 2005a, Gao *et al.* 2006] have been studied as graph-based descriptors. Based on the graph characterizations, segmentation can be made crossing conjunctions of graph branches, and each segmented sub-part has its sub-graph as internal structural characterization. In 2006, Biasotti *et al.* [2006] presented a Reeb graph method to address the sub-part correspondence issue. The Reeb graph segmentation and the internal sub-graphs are adopted as the correspondence signature. On the other hand, Xie *et al.* [2008] employed a skeleton based criterion to find out sub-part correspondences with in 2D plane domain. Bai *et al.* [2007] also presented a hierarchical skeleton graph representation for local feature segmentation and comparison based on Dilation Based Multi-resolutional Skeleton (DBMS) generation on 3D volumes. The DBMS based local feature descriptor can capture both geometric and topological features of the segmented sub-parts.

However, the above structural methods are sensitive to minor changes of 3D geometries, such as parametrically designed variants, scanned data noises, and polygonal mesh differences. More importantly, the aforementioned segmentation and partial matching schemes only consider geometric and topological aspects. As a consequence, the produced segmentation results, such as surface patches or shape fragments are "dump" surfaces or solids, which are of little value in the reuse stage. In addition, effective reuse of any segmented patch in the form of NURBS surfaces is still an open challenge [Zhao *et al.* 2009]. Therefore, partial shape retrieval on CAD model components ideally requires a segmentation method, which is insensitive to geometrical or topological changes. And the partial matching methods should have ability to analyze the semantic data available in feature-based CAD models. The usage of semantic data can produce mechanically meaningful segments, which pose desirable reusability after retrieval.

2.3 Summary

The purpose of this chapter is to describe the related research background in the field of 3D content-based retrieval. It presents a meaningful literature review of existing 3D matching methods based on generic and partial shape similarity, with an emphasis on retrieving CAD models. Meanwhile, some gaps existing in the current literature have been identified.

From the above review, the gap between retrieval and reuse is the major issue for effective reuse-oriented CAD model retrieval. Most existing methods of generic shape matching are rigid shape based, and they work on geometric representation only, *i.e.*, meshed or surface models. Retrieved results of such rigid shape matching are not easy to be manipulated, which causes the inflexibility to reuse after retrieval. Few methods consider the design reusability into the similarity assessment criteria. The same drawback is identified in existing partial shape matching methods, which have been supported by stochastic or clustering techniques. Both techniques only consider geometric and topological aspects, eventually producing mechanically meaningless retrieval results which are hard to reuse. It is therefore desirable to develop a reuse-oriented retrieval method that performs CAD model retrieval more effectively for better downstream design reuse. In the following chapters, two reuse-oriented CAD model retrieval methods based on respective essential and partial shape similarities will be proposed to achieve this.

Chapter 3   Knowledge Acquisition and Representation

This chapter presents a design knowledge based method to acquire mechanical modeling knowledge from CAD models and represent the expertise in a hierarchically structured representation, which will facilitate the next reuse-oriented retrieval.

In CAD domain, design is the means by which mechanical engineers can create manufactured parts using tools like computers. With the help of the CAD technology, a part can be represented as a digitalized computer model showing all the dimensions necessary to manufacture or assembly it. This digitalized design information makes it possible for designers to reuse the same knowledge to facilitate next designs as computers are suitable to store and replay massive data quickly.

However, design related expertise is complicated to be fully understood by computer programs themselves. In a mechanical design, there are enormous core concepts, including mechanics, kinematics, thermodynamics, materials science, structural analysis *etc*. In this thesis, the design information pertinent to parametric and feature based modeling (PFM) is extracted for the reuse-oriented retrieval. The rational is that nowadays PFM is a dominating technique to create mechanical CAD models. Therefore, the PFM knowledge embedded in these models should be utilized to facilitate mechanical re-design.

In this chapter, the definition and properties of the PFM design knowledge will be introduced. Then the chapter will present the method to extract PFM design knowledge from feature-based CAD models. The method how to normalize and represent the extracted knowledge in a structured way is also introduced, which enables the knowledge-based reuse-oriented retrieval algorithms.

3.1 Modeling Dependency between Features

In most CAD systems, the PFM technique is readily available today. PFM is popular not only because the created features are adjustable, which are easier to the end user, but also it provides the capabilities to embed the original designer's knowledge into the PFM-created model. The embedded knowledge enables the next designer to understand the original design intents. The knowledge preservation and transfer ensure that the future modifications can be done easily and consistently.

In PFM enabled modeling system, a CAD model is created using a series of design features. The design features have a semantically higher level than geometric primitives [Shah and Mäntylä 1995]. A design feature is defined as a parametric shape associated with design related attributes. The associated attributes include not only the information of itself, such as dimensions, orientations, tolerances, material properties, but also modeling references to other features [Mantyla *et al.* 1996]. Therefore, in PFM modeling, a feature will not be created alone, and the creation of one feature could depend on other features. For instance, an edge blend on the boundary of an extrusion is dependent on the extrusion. That means creation of the extrusion must be a preceding action before creating the edge blend. In the following section, the definition of feature modeling precedence will be given.

3.1.1 Feature modeling precedence relation

The precedence between the extrusion and the blend can be defined as follows. Given two features *f* and *g*, we say *g* depends on *f* if the creation of *g* is referred to *f* by geometric constraints [Bettig and Shah 2001]. In other words, *g* will be built after creating *f* due to the modeling reference to *f*.

The precedence defined in the aforementioned case is a partially ordered relationship, which can be denoted as:

$$f \rightarrow g \tag{3.1}$$

In the equation, the arrow direction indicates a sequencing of the partially ordered pair, *f* and *g*. In PFM modeling context, the sequencing is the modeling precedence of feature *f* over feature *g*. A more complex modeling precedence will be given as follows. Figure 3-1 shows design feature constitutes of the CAMI-ANC 101 part [Shah and Mäntylä 1995].

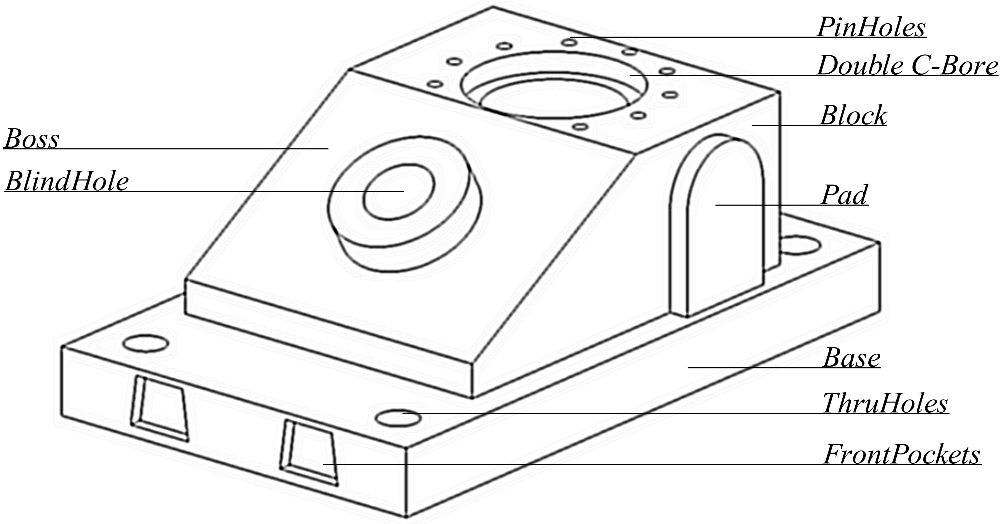


Figure 3-1. The ANC-101 model and its design features

In the CAMI-ANC 101 model, the feature *Pad* is build upon features *Base* and *Block* by coplanar constraints. Therefore, modeling precedence from *Base* and *Block* to *Pad* will be many-to-one. Here, this many-to-one relationship is decomposed into two partially ordered pairs, say *Base*→*Pad* and *Block*→*Pad*, each of which represents the modeling precedence of *Base* over *Pad* and *Block* over *Pad*, respectively. The research annotates this decomposition as the normalization of multi-precedence.

In the similar way, another one-to-many precedence relationship, *i.e.*, *Block* is also the direct modeling reference for feature *Boss*, *Pin-Holes* and *Double C-Bore*. Consequently, three discrete partially ordered pairs are defined as *Block*→*Boss*, *Block*→*Pin-Holes*, and *Block*→*Double C-Bore*, which use the directed arrows to show modeling precedence between these features.

3.1.2 Properties of feature modeling precedence

The partially ordered relationship defined in section 3.1.1 satisfies these properties: *irreflexivity*, *antisymmetry*, and *transitivity*. The irreflexivity property prevents a feature from depending on itself; while the antisymmetry property ensures that two features must not mutually dependent on each other. Moreover, the transitivity property is satisfied as f will be the indirect reference for generating h if $f \rightarrow g$ and $g \rightarrow h$ in feature mechanical modeling. These three properties are formalized as follows.

$$\begin{cases} \text{Irreflexivity}: \forall a \in F(M), \neg(a \to a) \\ \text{Asymmetry}: \forall a,b \in F(M), a \to b \Rightarrow \neg(b \to a) \\ \text{Transitivity}: \forall a,b,c \in F(M), a \to b \wedge b \to c \Rightarrow a \mapsto c \end{cases} \tag{3.2}$$

For the feature set $F(M)$ of a CAD model M, the following properties apply to all the features a, b, or c in $F(M)$. The notation $\mapsto$ stands for a transitive precedence. Since the irreflexivity, asymmetry, and transitivity properties apply to the feature modeling precedence, it is a strict partially ordered relation in order theory.

3.2 Acquisition of Feature Modeling Precedence

A PFM-created CAD model maintains two distinct representations. The geometric representation describes the shape of the CAD model in terms of B-Rep entities, and another feature representation includes design features and their relationships. The proposed method will extract the feature modeling semantics from the PFM-created model.

The expertise to be extracted includes design feature constitutes and the modeling dependencies among them. The design feature information is extracted from the design history of the PFM-created models, where the names, geometries, and properties of the features can be obtained. Due to the use of feature information, the proposed method restricts its comparable objects to feature modeled CAD models. The rationale is that the PFM technique is acutely evident in current mechanical parts and assembly designs. Moreover, the PFM-created CAD model is feasible for variant and adaptive redesign for design reuse.

The modeling precedence between features can be deduced from the inter-relationship between two features. For instance, if deletion of feature *F* causes removal of another feature *G* from the PFM-created CAD model, *G* has at least one modeling dependency on *F*. Therefore, every modeling precedence relation can be computed by the transversal of all design feature pairs in a PFM-created CAD model. The modeling precedence computed is a strict partially ordered relation between the corresponding design feature pairs; and these pairs of precedence put together will form a strict partially ordered set.

3.3 Representation of Modeling Precedence Knowledge

The aforementioned strict partially ordered theory provides a theoretic foundation to represent complicated feature modeling precedence in a structured way. In the order theory [Schröder 2002], a strict partially ordered set (POSET), such as the feature constitutes of a CAD model, can be represented by a directed acyclic graph.

3.3.1 Directed acyclic graph

A directed acyclic graph (DAG) is a directed graph with no directed cycles. It is formed by a set of nodes and a set of edges connecting pairs of vertices. The acyclic property ensures that there is no way to start at one vertex *v* and follow a sequence of directed edges that eventually loops back to *v* again.

In a directed acyclic graph $G= (V, E)$, where *V* is the *vertex* set and *E* is the directed *edge* set. An edge $e = (v, w)$ is considered to be directed from vertex *v* to another vertex *w*. *w* is said to be a direct successor of *v*, and *v* is said to be a direct predecessor of w. If a *path* consisting of one or more successive edge leads from v to w, then w is said to be a successor of v, and v is said to be a predecessor of w.

A chain of successors forms a reachable path from a root to the rest of the vertex. If there is a path is started at vertex u and ended at vertex v, we say that v is reachable from u. In graph theory, *reachability* is the term to describe the relation that gets from one vertex in a directed graph to some other vertex. The reachability is defined as follows. For a directed graph $G= (V, E)$, the reachability relation of G is the transitive closure of its directed edge set E, which is to say the set of all ordered pairs (v, w) of vertices in V for which there exist vertices $v_0 = v, v_1, \ldots, v_n = w$ such that (v_i, v_{i+1}) is in E for all $0 \leq i < n$.

The defined reachability is used throughout the thesis to prescribe the non-immediate modeling precedence relation of feature constitutes. In feature modeling context, all successors in the reachable path of v means these successors have immediate or transitive modeling dependency on v.

3.3.2 Feature directed acyclic graph (FDAG)

The defined DAG is used to model partially ordered, feature modeling precedence in the proposed reuse-oriented retrieval method. This graph-based model is named as Feature Directed Acyclic Graph (FDAG). In an FDAG model, the vertices are design feature constitutes of a PFM-created CAD model; the edges of the DAG represent modeling precedence between design features.

An FDAG representation of a PFM-created CAD model *M* is constructed as follows:

(1) Construction of vertices. For each design feature visited $\{f_1, f_2, \ldots, f_n\}$, vertices $\{v_{f1}, v_{f2}, \ldots, v_{fn}\}$ will be put into an empty directed acyclic graph (called *G*) correspondingly.

(2) Construction of edges. Traverse the vertices $\{v_{f1}, v_{f2}, \ldots, v_{fn}\}$ of *G* in a pairwise way. When a pair $<v_{fi}, v_{fj}>$ is visited, a directed edge is inserted from v_{fi} to v_{fj} if f_i is a directly (not transitively) preceding design reference to f_j in the model.

Once all vertex pairs are visited, and the corresponding directed edges are inserted, the FDAG graph of the input CAD model has been constructed. The FDAG of the ANC-101 part is illustrated in Figure 3-2.

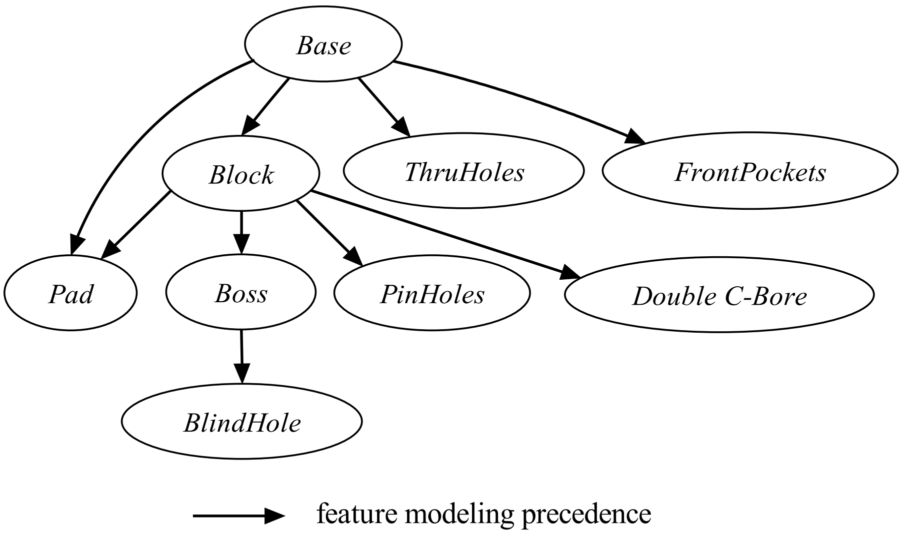


Figure 3-2. FDAG graph of the ANC-101 part shown in Figure 3-1

It is noted that the FDAG graph is not equivalent to the design history. A design history records the modeling process defined by designers. As the term *history* suggests, all feature constitutes present in a model are recorded chronologically. A chronological order indicates a *totally ordered set*, where any pair of feature constitutes are comparable by their instantiation times. Dissimilarly to the totally ordered design history, the feature modeling precedence defined in the FDAG graph is a partially ordered relation, which means that only the pair having the precedence relation is comparable. In other words, a predecessor feature in the design history is not necessary to be a successor in modeling precedence, because there could be no geometric constraints between them at all. Therefore, the FDAG representation proposed in this research is not equal to the user defined design history.

The differential ordering properties reveal marked difference between design history and FDAG. As the design history is a totally ordered set, in which a single design change would create a largely different design history. However, the FDAG graph is a partially ordered set, which only records the necessary precedence if it exists. Because of the top-down thinking of the production design in human beings, gradual modeling practices from coarse-to-fine are quite common in design activities, especially for a design is started from scratch. Therefore, compared with variably created design histories, the modeling precedence can be relatively invariable because a principle has to be followed - “minor details are always built upon major components”.

Taking the ANC-101 part as an example, there are two kinds of possible design histories as illustrated in Figure 3-3, which are alternatives to build the part. This figure clearly shows that design histories may differ from person to person: one designer might create *ThruHoles* in the very first, while another user would like to add it into the model in a later stage. These varieties exist in a totally ordered set. However, the variables illustrated in Figure 3-3 correspond to an invariable modeling precedence relationship, as shown in the partially ordered set (*i.e.*, the FDAG graph of the Figure 3-2). This observation confirms that the proposed FDAG graph, which is based on partially ordered set, is able to capture relatively invariable essential of feature modeled CAD models.

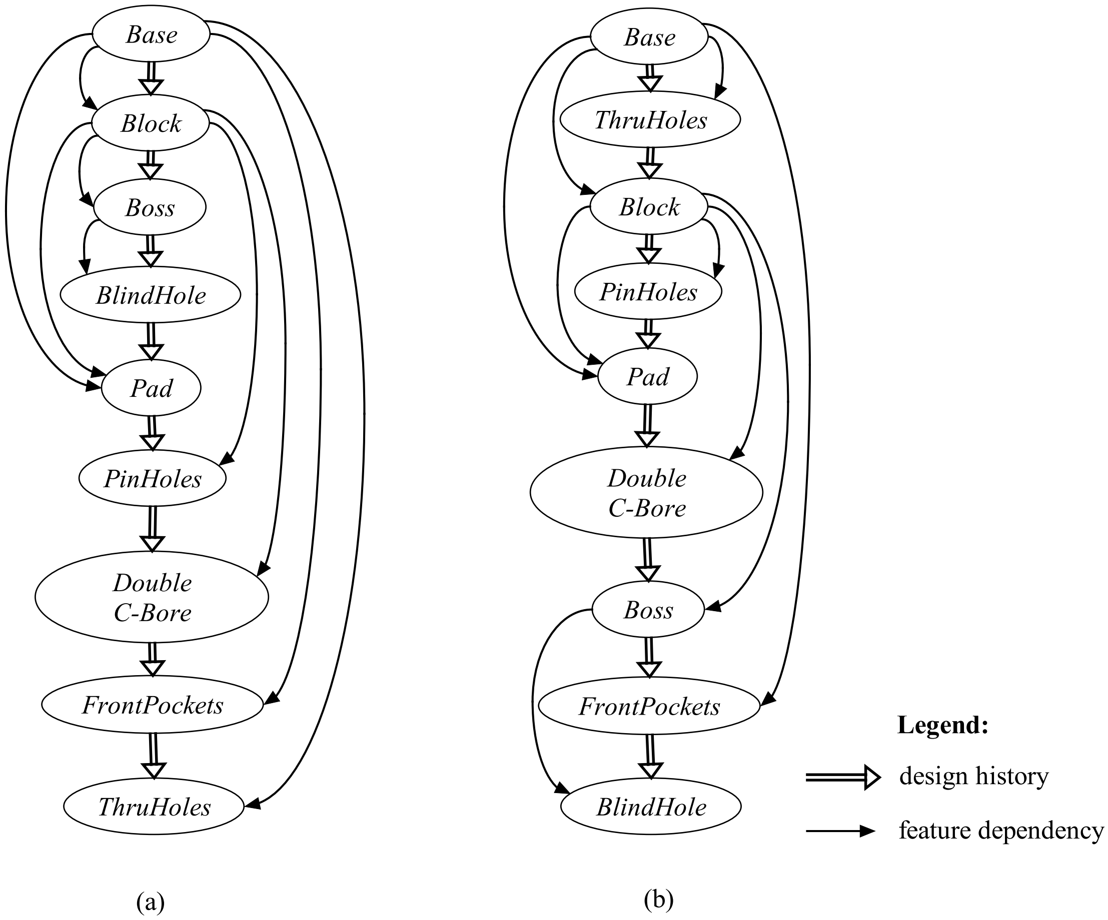


Figure 3-3. Two design history alternatives for ANC-101 part

The FDAG graph proposed in this chapter not only provides a hierarchically structured, easy to understand data structure for representing embedded knowledge of CAD models, but also accurately acquires relatively invariable modeling precedence essential instead of volatile design histories. The modeling precedence acquired will be utilized to analyze the component dependency of mechanical parts, and further support the CAD model retrieval based on general or partial shape similarity.

Chapter 4 Retrieval Based on Essential Shape Similarity

Chapter 3 has introduced a modeling knowledge acquisition and representation mechanism, which extracts partially ordered precedence from feature constitutes of CAD models for the reuse-oriented retrieval. All extracted partially ordered relations constitute a partially ordered set (POSET), which exactly corresponds to an acyclic graph structure. The graph can be equivalently transformed to a hierarchically structured knowledge representation. In this chapter, this hierarchical graph structure, namely Feature Directed Acyclic Graph (FDAG), will be used to implement CAD model analysis and similarity assessment for design reuse.

Currently, 3D CAD model retrieval methods do not support design reuse well, especially with complicated mechanical designs (see Figure 4-1) and even more complicated modeling dependencies. In order to facilitate CAD model reuse activities, chapter 4 and chapter 5 describe two novel reuse-oriented retrieval approaches for essential and partial similarity assessment. In the approaches to be presented, a series of CAD model decomposition, *i.e.*, essential component simplification and sub-part segmentation will be applied on complex feature modeled CAD models. The decomposed models are characterized and compared in the similarity and reusability assessment. Consequently, users are able to retrieve CAD model based on essential or partial shapes. Moreover, the decomposition information can be utilized as re-design reference to facilitate the design reuse stage. Therefore, design reuse activities are supported by the proposed CAD model retrieval algorithms, from beginning to end.

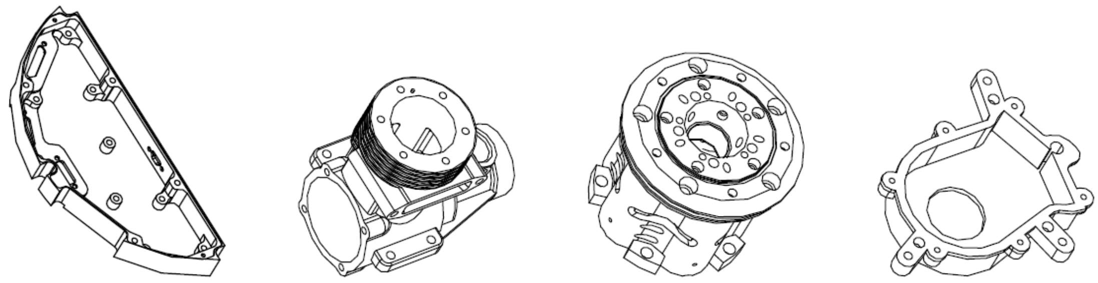


Figure 4-1. Examples of mechanical parts [Bespalov *et al.* 2005]

An ideal reuse-oriented retrieval method is to match reusable CAD models based on the essential shape similarity between the user-specified query and parts in repositories. This kind of CAD model retrieval comes from a common reuse requirement: designers want to search for similar parts that were completed in the past for redesign reference. However, after a long time, only significant shape of past designs can be easily recalled. Therefore, using a query to fetch all previous parts that have similar overall shapes will be a necessity in the reuse-oriented retrieval paradigm. This retrieval on essential shape similarity also benefits the design reuse stage because design details normally are required to be adjusted according to differential design requirements. CAD model matching on the general similarity will return designers the most reusable major shapes, irrespective of trivial details attached. This chapter will describe this essential shape retrieval method that is shown in Figure 4-2.

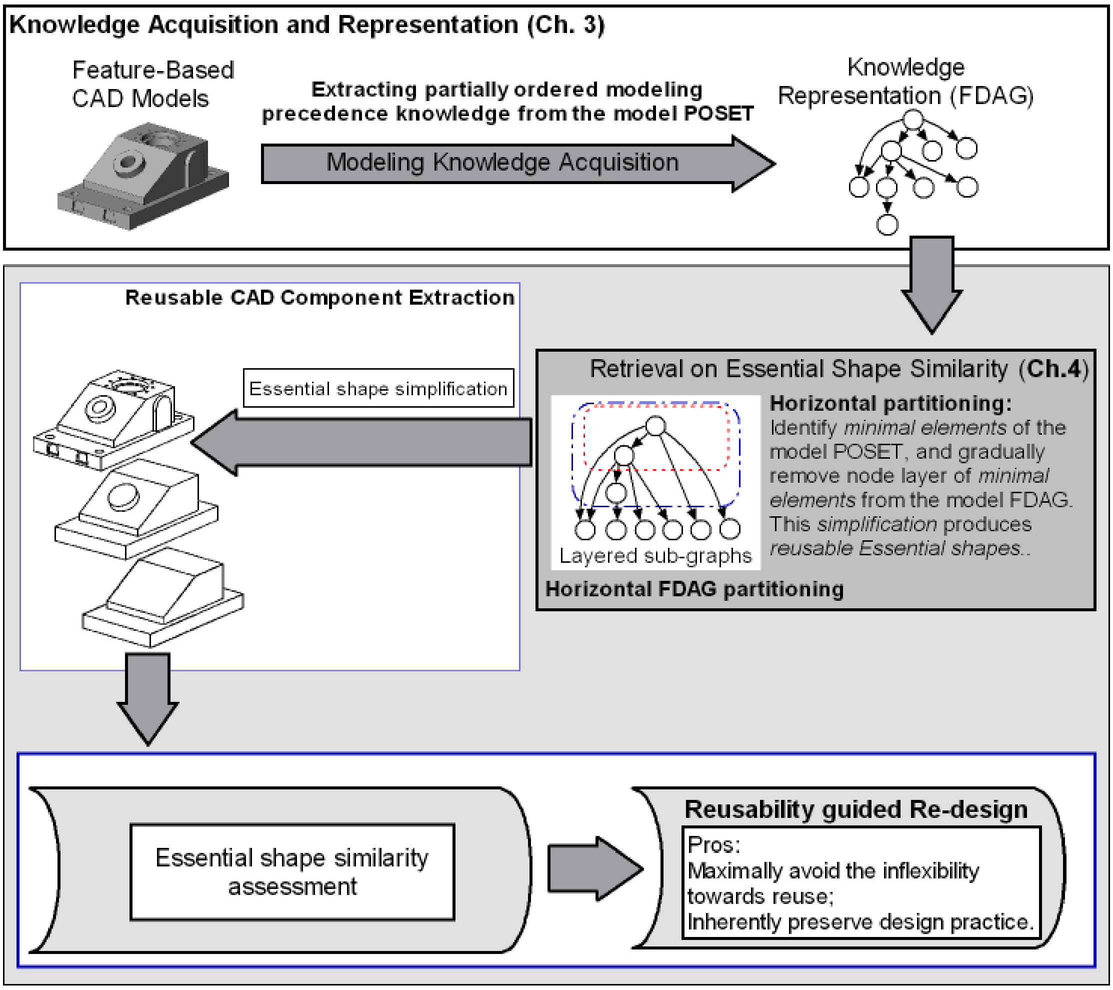


Figure 4-2. Flow chart of the essential shape retrieval method

4.1 Essential Shape Retrieval

Real-world CAD models have highly complicated structures satisfying requirements from functional, assembly and structural aspects. Most research works on 3D shape retrieval focus on the mesh or surface representation. These methods are sensitive to geometrical details; therefore, they can barely obtain satisfactory results on realistic-scale CAD models due to the complexity of their elaborate shape details.

In order to address the aforementioned obstacles and retrieve CAD models based on their overall shapes, an essential shape similarity assessment method is proposed in this chapter. The method progressively simplifies CAD models from fully detailed to less detailed, according to a horizontal FDAG graph partitioning algorithm proposed in following sub-sections. All simplified models are indexed into a concise representation, which is named as *essential shape aggregation* (ESA) descriptor. With the help of the ESA descriptor, essential similarity assessment can be done to search complicated CAD models, using some essential shapes as query models. As the semantic-driven CAD model simplification is conforming to original modeling dependencies, the proposed essential shape matching algorithm will significantly facilitate the design reuse as the reuse inflexibility would be maximally avoided. In addition, design intents of previous designers can be inherently preserved during the retrieval and reuse.

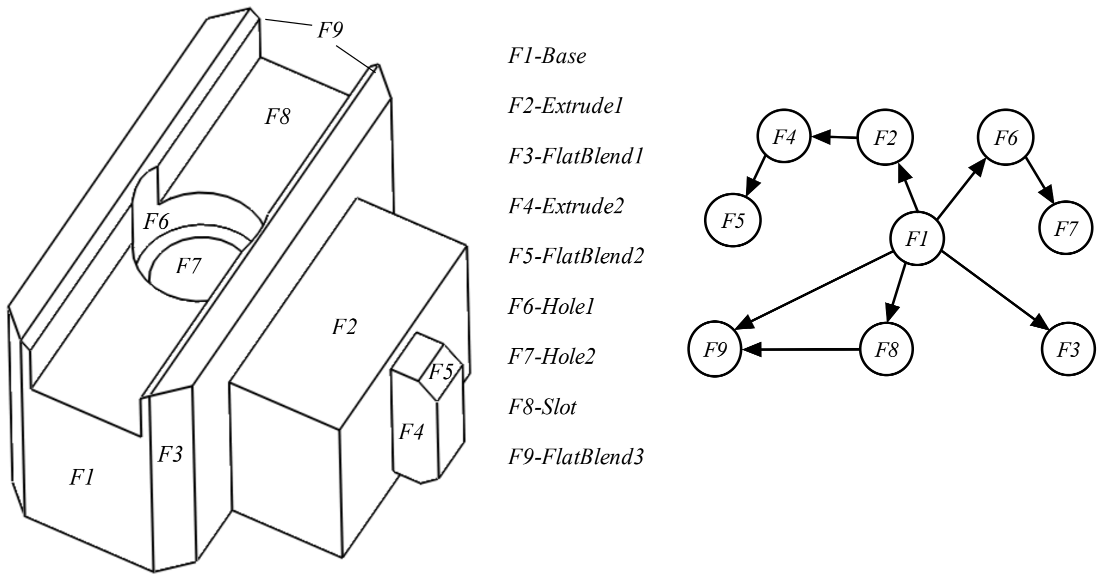
4.2 Knowledge-Based Horizontal Partitioning

The domain knowledge that is used to develop essential shape matching algorithm is the feature modeling precedence introduced in Chapter 3. The modeling precedence is a partially ordered relation, and each individual dependency forms part of a complete partially ordered set, which is corresponding to the whole CAD model.

Given a pusher pad mechanical model as an example shown in Figure 4-3a, it consists of 9 modeling features. These features can be constructed in various design history sequences (the sign > here is the totally ordered chronological sequence), *e.g.*,

i. F1 > F3 > F2 > F4 > F5 > F8 > F9 > F6 > F7

ii. F1 > F8 > F9 > F6 > F7 > F2 > F4 > F5 > F3



(a) The model (b) The FDAG graph

Figure 4-3. The feature-based pusher-pad model and its FDAG graph

No matter which sequence is used in final design, the following critical modeling precedence must be followed: {*F1* > *F2* >*F4* >*F5*}, {*F1* > *F3*}, {*F1* > *F8*}, {*F8* > *F9*}, {*F1* > *F6* > *F7*}. In fact these precedence sub-sequences correspond to the feature POSET of this pusher-pad model. These indispensable precedence relations can be visualized by its FDAG graph shown in Figure 4-3b. Due to the properties of partially ordered set, there is no loop in the graph.

In order theory, for a vertex v, the number of direct predecessors adjacent to it is called the *in-degree* of it and the number of direct successors is its *out-degree*. The *in-degree* is denoted $deg^{-}(v)$ and the *out-degree* as $deg^{+}(v)$. There are elements having extreme *in-degree* and *out-degree*. They are *maximal* and *minimal* elements, notably:

- *Maximal element*: An element a in a POSET S is a maximal element if there is no element b in S such that $b \rightarrow a$ in terms of the partial precedence. In a directed graph, in-degree of each maximal element will be zero. A node v is a maximal element if and only if $deg^{-}(v)$ equals to zero, *i.e.*

$$v \text{ is a maximal element of a DAG} \Leftrightarrow \deg^{-}(v) = 0 \qquad (4.1)$$

- *Minimal element*: Similarly, an element c in the POSET S is a minimal element if there is no other element d in S such that $c \rightarrow d$. In a directed graph, the minimal element v must be an element whose $deg^{+}(v)$ is zero.

$$v \text{ is a minimal element of a DAG} \Leftrightarrow \deg^{+}(v) = 0 \qquad (4.2)$$

Based on the definitions (4.1) and (4.2), the vertices of any POSET can be categorized into three groups: maximal elements, minimal elements, and normal ones which do not fall into any of the above two groups. It can be proven that in every acyclic directed graph G there is at least one vertex with zero in-degree (*i.e.*, maximal element) and at least one vertex with zero out-degree (*i.e.*, minimal element) [Deo 1974]. A close observation at the FDAG shown in Figure 4-3b reveals that the maximal element group has only one vertex {*F1*}, and that the minimal element group has four members {*F3*, *F5*, *F7*, *F9*}. In feature-based modeling context, those maximal elements represent feature constitutes normally having most modeling significance. In CAD modeling, all others features are built upon the most significant constitute, *e.g.*, *F1 (Base)* block in the pusher-pad model (Figure 4-3a). Conversely, the minimal element set containing feature constitutes has less modeling significance.

Using the mechanical knowledge-based categorization of maximal and minimal elements of feature modeling constitutes, a re-organization can be made on the original FDAG. In the re-organization, the maximal elements are placed in the top, and the minimal elements are arranged in the bottom. For example, the FDAG shown in Figure 4-3b is re-organized into the one shown in Figure 4-4.

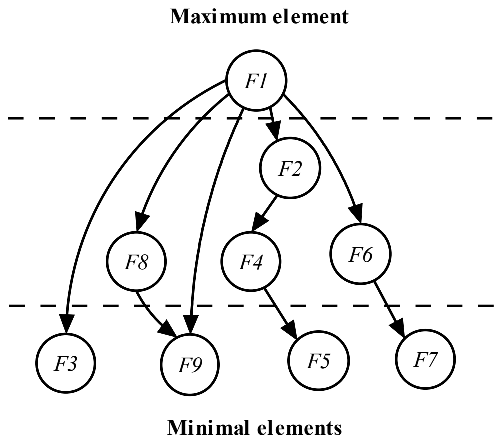


Figure 4-4. The re-organized FDAG graph from the FDAG shown in Figure 4-3b

Figure 4-4 clearly shows a hierarchically ordered representation, in which all directed edges have a natural top-down orientation, from the maximum elements towards the minimum ones. These top-down orientations exactly correspond to the feature modeling precedence sequences between vertices of the FDGA. Based on this knowledge-based, hierarchical representation, we are able to analyze the simplification of complex feature modeled CAD models. The conversion process from a raw FDAG graph to this kind of hierarchically ordered representation is named as the *FDAG normalization*.

After normalization, FDAG nodes in the bottom are all minimal elements without children, *i.e.*, no succeeding nodes in terms of feature modeling precedence. Since feature modeling precedence relation is asymmetric, as referring to Eq. (3.2), for any leaf node g and its preceding vertex f, deleting f leads to the removal of g, but not vice versa. On the other hand, g can be removed without affecting f because

$\neg(g \rightarrow f)$. Therefore, removing this kind of minimal elements from the FDAG will not violate any existing feature modeling precedence. For example, after the removal of minimal elements {*F3*, *F9*, *F5*, *F7*} from the FDAG shown in Figure 4-4, the simplified result is still a valid model, just being more general without a few decorating trivial features, as shown in Figure 4-5.

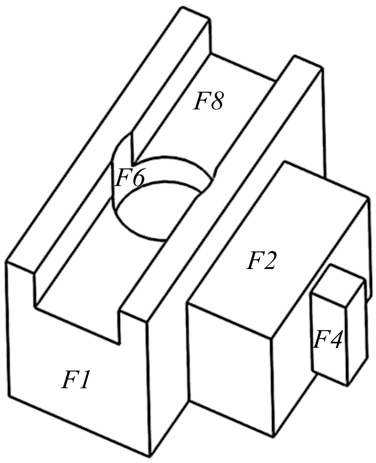


Figure 4-5. The simplified pusher-pad model after removing minimal elements from the corresponding FDAG.

Such removal of minimal elements, without affecting modeling precedence relations of other vertices, is defined as a *knowledge-based simplification* of the CAD model. Considering the removed FDAG minimal elements {*F3*, *F5*, *F7*}, they are corresponding to minor features, such as flat edge blends. The main reason of the small volume of these minimal elements is that designers embody their designs in a coarse-to-fine way, and minor features are always built upon major components. Based on these analyses, we can deduce the geometric significance of feature constitutes from the FDAG representation. Therefore, the normalized FDAG hierarchy can effectively reveal geometric significance of modeling features, based on the above-mentioned knowledge reasoning.

From graph theory perspective, every round of simplification corresponds to a horizontal partitioning that splits leaf nodes from the normalized FDAG, which can be observed from the example shown in Figure 4-5. After the simplification of the pusher-pad model, another three nodes become the new minimal elements. They are {*F8*, *F4*, *F6*} as shown in Figure 4-6.

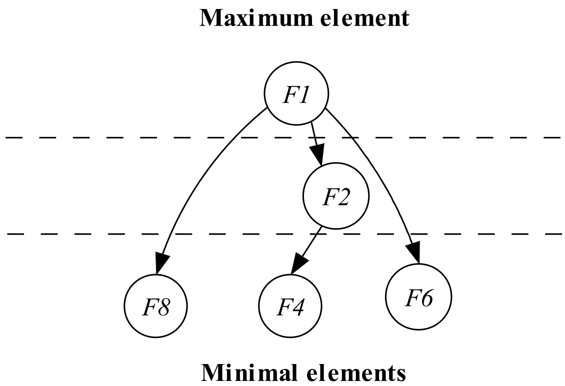


Figure 4-6. New minimal FDAG elements after one round of simplification on the pusher-pad model

Based on the same procedure, the model can be recursively simplified further by removing new minimal elements. In the meanwhile, the corresponding CAD model is partitioned from full detailed to less detailed. This recursive procedure is named as multi-level simplification of CAD models, which can effectively simplify complex CAD models without violating existing modeling dependencies. The algorithmic details of the multi-level simplification will be given in the next section.

4.3 Multi-Level Simplification of CAD Models

Because of the asymmetric modeling precedence, all minimal elements of the FDAG graph can be removed in a batch. From the feature modeling perspective, the batch removal is a simplification of insignificant feature constitutes in terms of feature modeling precedence, without violating any existing modeling dependencies. From another viewpoint of the graph, this removal is exactly mapping to a horizontal partitioning of leaf nodes from the complete FDAG graph. More importantly, the horizontal partitioning can be recursively applied on the partitioned FDAG graph to remove more insignificant feature constitutes from the original CAD model. This recursive procedure is defined as the *multi-level simplification* of CAD models, a key process of essential shape retrieval algorithm.

The basic idea of multi-level simplification is to remove leaf nodes from a complete FDAG graph layer by layer, without affecting adjacencies of other vertices. By keeping the horizontal partitioning, a fully detailed CAD model is progressively simplified into less detailed ones. With details removed, essential shapes of CAD models can be obtained. The steps of the proposed multi-level simplification are listed as follows:

(1) Traverse vertices of FDAG G of the CAD model M. For every vertex visited, mark it if the out-degree is zero. A nil out-degree indicates that the vertex is a minimal element of the FDAG and thus needs to be removed;

(2) Traverse edges of the FDAG. Remove all edges pointing to the marked vertices;

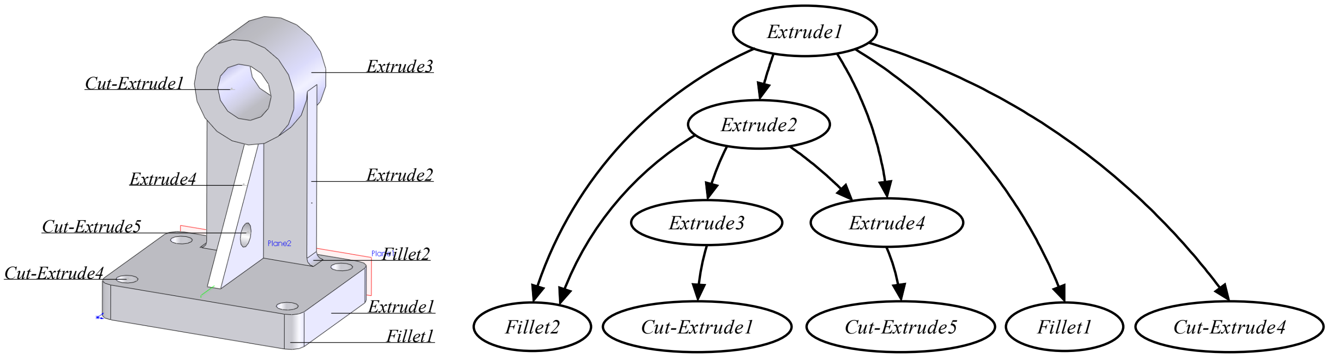
(3) Delete all marked vertices. Remaining sub-graph G_i corresponds to a simplified shape M_i of the complete model M (i increments by 1 after every simplification).

(4) The algorithm returns to the step (1) until a pre-defined termination criterion is reached.

The pre-defined termination criterion is used to prevent an original CAD model from being changed too much. The criterion is a threshold of the bounding box discrepancy between the original and a simplified shape. In this research, the bounding box of a CAD model is computed by the axis-aligned method, where boxes are aligned with the axes of the world coordinate system (WCS) of the model. The choice of axis-aligned bounding box (AABB) is due to the fact that the CAD model is hardly rotated during the design stage, and the AABB calculation is simple.

If $\{M_0, M_1, \ldots, M_i, \ldots, M_n\}$ represent the original model M_0 and its n simplified shapes, $AABB(M_i)$ stands for the axis-aligned bounding box of M_i, and $\Delta AABB(M_i)$ indicates the bounding box discrepancy between M_0 and M_i (i = 1 to n). For any simplification, $|\Delta AABB(M_i)| \leq \delta \times AABB(M_0)$ should be satisfied. Based on our experiments, $\delta = 0.25$ works well in all cases.

The simplification is completely guided by vertex properties of FDAG; therefore it can also be automatically executed by a computerized algorithm. The algorithm is illustrated by the example shown in Figure 4-7. Figure 4-7 shows that (a) is the feature constitutes of the part and (b) the normalized FDAG graph of the part.



(a) The model (b) The FDAG

Figure 4-7. A feature-based model of a bracket part and its normalized FDAG

In mathematics, an adjacency matrix is a means of representing which vertices of a graph are adjacent to which other vertices. An adjacency matrix A of an FDAG $G = (V, E)$ is a matrix $[a_{ij}]_{n \times n}$ where n is the total count of elements in the FDAG. The value a_{ij} indicates the adjacency between the vertex v_i and v_j ($v_i, v_j \in V$) as:

$$a_{ij} = \begin{cases} 1, & v_i \rightarrow v_j \\ 0, & \text{othewise} \end{cases} \tag{4.3}$$

Because the FDAG is a finite acyclic rooted graph, either a depth-first search (DFS, visiting child nodes before sibling nodes) or breadth-first search (BFS, visiting sibling nodes before child nodes) can be used. Element sequence of an FDAG adjacency matrix is defined as the sequence of graph traversal which begins with the rooted node. In this research, DFS is adopted for graph traversal. If DFS is applied to the normalized FDAG in Figure 4-7b, the sequence is *Extrude1*, *Fillet2*, *Extrude2*, *Extrude3*, *Cut-Extrude1*, *Extrude4*, *Cut-Extrude5*, *Fillet1*, and *Cut-Extrude4*. Using this sequence for rows and columns, a readable adjacency matrix of it is shown in Table 4-1. The dimension of the matrix is equal to the number of features of the part as well as the vertex count in the FDAG. Based on the definition in the Equation

(4.3), the value 1 of $a_{i=3,\, j=4}$ indicates that there is a directed edge leaving from $v_{i=3}$ (*Extrude2*) towards $v_{j=4}$ (*Extrude3*).

Table 4-1. The FDAG adjacency matrix of the bracket

v_i \ v_j		*j=1*	*j=2*	*j=3*	*j=4*	*j=5*	*j=6*	*j=7*	*j=8*	*j=9*
		Extrude1	*Fillet2*	*Extrude2*	*Extrude3*	*Cut-Extrude1*	*Extrude4*	*Cut-Extrude5*	*Fillet1*	*Cut-Extrude4*
i=1	*Extrude1*	0	1	1	0	0	1	0	1	1
i=2	*Fillet2*	0	0	0	0	0	0	0	0	0
i=3	*Extrude2*	0	1	0	1	0	1	0	0	0
i=4	*Extrude3*	0	0	0	0	1	0	0	0	0
i=5	*Cut-Extrude1*	0	0	0	0	0	0	0	0	0
i=6	*Extrude4*	0	0	0	0	0	0	1	0	0
i=7	*Cut-Extrude5*	0	0	0	0	0	0	0	0	0
i=8	*Fillet1*	0	0	0	0	0	0	0	0	0
i=9	*Cut-Extrude4*	0	0	0	0	0	0	0	0	0

Using the adjacency matrix shown in Table 4-1, the out-degree of every FDAG vertex $deg^{+}(v_i)$ is calculated by accumulating all values of $a_{i,\, j=1,\ldots,n}$, as:

$$deg^{+}(v_i) = \sum_{j=1}^{n} a_{ij} \tag{4.4}$$

Similarly, the in-degree, $deg^{-}(v_i)$ can be obtained by accumulating vertical values of certain column as:

$$deg^{-}(v_i) = \sum_{j=1}^{n} a_{ij} \tag{4.5}$$

All FDAG nodes are listed on the left-most column of Table 4-2, and using data in Table 4-1 and Equation (4.4), the corresponding out-degree of each node is listed in the second column. A total of five vertices are found to be the minimal elements.

They are *Fillet2*, *Cut-Extrude1*, *Cut-Extrude5*, *Fillet1*, and *Cut-Extrude4*, which are marked with (*) in Table 4-2.

Table 4-2. Out-degree of FDAG vertices during the multi-level simplification

FDAG vertices *V*	**Original Model**	**After first level simplification**	**After second level simplification**
***v** traversed by DFS*	$\boldsymbol{deg^{+}(v)}$	$\boldsymbol{deg^{+}(v)}$	$\boldsymbol{deg^{+}(v)}$
Extrude1	5	2	1
Fillet2	0 (*)	N.A.	N.A.
Extrude2	3	2	0 (#, the termination criterion is reached)
Extrude3	1	0 (*)	N.A.
Cut-Extrude1	0 (*)	N.A.	N.A.
Extrude4	1	0 (*)	N.A.
Cut-Extrude5	0 (*)	N.A.	N.A.
Fillet1	0 (*)	N.A.	N.A.
Cut-Extrude4	0 (*)	N.A.	N.A.

According to the multi-level simplification algorithm, the found minimal elements and their connected edges will be removed from the FDAG. The first round of simplification will purge five minimal elements, *Fillet2*, *Cut-Extrude1*, *Cut-Extrude5*, *Fillet1*, and *Cut-Extrude4*. From the viewpoint of graph theory, these features are all constitutes upon which no other are constructed. From human perception, these are all minor features, such as corner edge blends and locating holes.

After the first round of simplification, the multi-level algorithm recursively traverses the purged DFAG again to choose minimal elements for next removal, if the defined termination criterion is not reached yet. As shown in the third column of Table 4-2, the out-degree of *Extrude3* and *Extrude4* is nil, and therefore they are marked as leaf nodes and will be simplified in this round. After the second round simplification, the remaining CAD model still has *Extrude1*and *Extrude2*, and *Extrude2* becomes the new minimal element. The simplification will stop until the pre-defined termination criterion is reached.

The same simplification process is also illustrated by the example shown in Figure 4-8, where FDAD leaf nodes are removed layer by layer, as illustrated by the dash lines. It can also be observed that from left to right, the resultant bracket part is progressively simplified from fully detailed to less detailed.

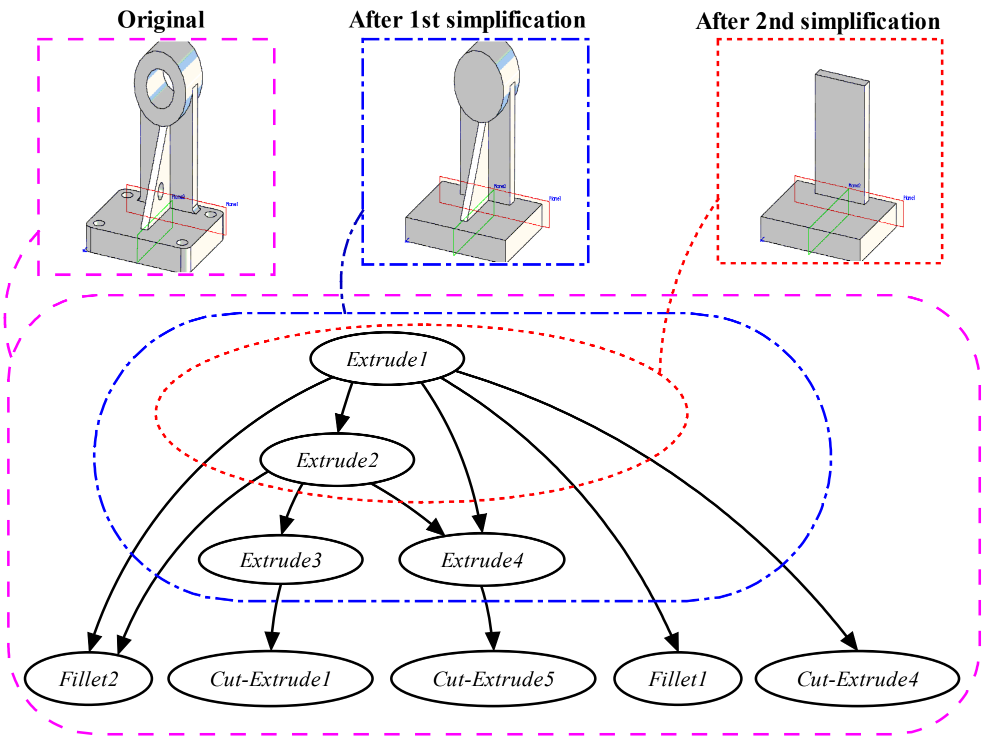


Figure 4-8. Multi-level simplification of the part shown in Figure 4-7

Most importantly, every simplified CAD model is valid in terms of feature modeling, throughout the FDAG simplification. All mandatory modeling precedence of remaining features will be kept unchanged because the knowledge-based simplification is only applied on those minimal elements from FDAG perspective. It can be clearly shown in Table 4-3 that during the same multi-level simplification process of Table 4-2, there are no changes on in-degree $deg^{-}(v)$ of any remaining node v, which exactly indicates the total number of mandatory modeling dependencies to build v.

Table 4-3. In-degree of FDAG vertices during the multi-level simplification

FDAG vertices *V*	**Original Model**	**After first level simplification**	**After second level simplification**
v	$\boldsymbol{deg^{-}(v)}$	$\boldsymbol{deg^{-}(v)}$	$\boldsymbol{deg^{-}(v)}$
Extrude1	0	0	0
Fillet2	2	N.A.	N.A.
Extrude2	1	1	1
Extrude3	1	1	N.A.
Cut-Extrude1	1	N.A.	N.A.
Extrude4	2	2	N.A.
Cut-Extrude5	1	N.A.	N.A.
Fillet1	1	N.A.	N.A.
Cut-Extrude4	1	N.A.	N.A.

The multi-level simplification described above has been implemented by an efficient algorithm based on linear transformation. In the algorithm implementation, an adjacency matrix representation is used, which is equivalent to the readable version of Table 4-1. The sequence of row and column in the matrix is defined by DFS sequence of FDAG graph traversal. The matrix representation of the original FDAG in Figure 4-7 is shown as follows (due to the irreflexivity property of FDAG nodes, the diagonal elements of the matrix are all zero):

$$\mathbf{A}^0 = \begin{pmatrix} 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

In each round of simplification, the found minimal elements and any directed edges pointing to them are to be removed. This linear transformation can be archived by multiplying the current adjacency matrix $\mathbf{A}^i$ with diagonal matrix $\mathbf{D}^i$ to remove the minimal elements from the original FDAG as $\mathbf{A}^{i+1} = \mathbf{A}^i \times \mathbf{D}^i$, where i is the number of linear transformation applied. In this diagonal matrix $\mathbf{D}^i$, all entries outside the main diagonal are zero. But on the diagonal line, the elements from Boolean domain $B = \{0, 1\}$, where the diagonal node d_{jj} is 0 if and only if the v_j is an FDAG minimal element, according to the Eq. (4.4). The diagonal matrix $\mathbf{D}^i$ is defined as:

$$\mathbf{D}^i = \begin{pmatrix} d_{11} & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & d_{jj} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & d_{nn} \end{pmatrix}, \text{ where } d_{jj} = \begin{cases} 0, & \text{if } v_j \text{ is minimal element} \\ 1, & \text{otherwise} \end{cases} \tag{4.6}$$

Combined with the minimal element's definition in Eq. (4.1) and the out-degree's definition in Eq. (4.4), Eq.(4.6) can be elaborated into:

$$\mathbf{D}^i = \begin{pmatrix} d_{11} & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & d_{jj} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & d_{nn} \end{pmatrix}, \text{ where } d_{jj} = \begin{cases} 0, & \text{if } \sum_{k=1}^{n} a_{jk} = 0,\ a_{jk} \in \mathbf{A}^i \\ 1, & \text{if } \sum_{k=1}^{n} a_{jk} \neq 0,\ a_{jk} \in \mathbf{A}^i \end{cases} \tag{4.7}$$

Taking $\mathbf{A}^0$ as an input, the diagonal matrix $\mathbf{D}^0$ is given as:

$$\mathbf{D}^0 = \begin{pmatrix} 1 &&&&&&&& \\ & 0 &&&&&&& \\ && 1 &&&&&& \\ &&& 1 &&&&& \\ &&&& 0 &&&& \\ &&&&& 1 &&& \\ &&&&&& 0 && \\ &&&&&&& 0 & \\ &&&&&&&& 0 \end{pmatrix}$$

A linear transformation can be applied to compute $\mathbf{A}^1$ by computing the cross product between $\mathbf{A}^0$ and the defined diagonal matrix $\mathbf{D}^0$. $\mathbf{A}^1$ is calculated as:

$$\mathbf{A}^1 = \mathbf{A}^0 \times \mathbf{D}^0 = \begin{pmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Compared with $\mathbf{A}^0$, $\mathbf{A}^1$ has the following properties. First of all, in $\mathbf{A}^1$, every element in columns {2, 5, 7, 8, 9} has been mapped to zero compared with $\mathbf{A}^0$, after the linear transformation. It means that all edges pointing to the current minimal elements have been removed during the matrix transformation. Secondly, there is no new directed edge added, which indicates that no feature modeling precedence is created in this transformation.

Figure 4-9 displays a directed graph corresponding to the matrix $\mathbf{A}^1$. It shows that the computation of $\mathbf{A}^1$ is exactly equivalent to the result of the first FDAG simplification, as illustrated in Figure 4-8.

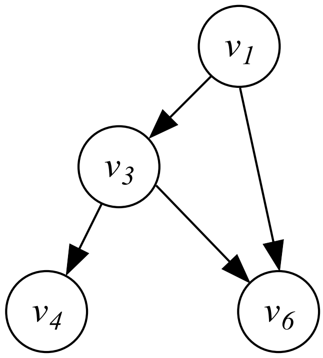


Figure 4-9. The directed graph corresponding to the adjacency matrix $\mathbf{A}^1$

Based on Eq. (4.7), $\mathbf{D}^1$ is computed as:

$$\mathbf{D}^1 = \begin{pmatrix} 1 & & & & & & & & \\ & 0 & & & & & & & \\ & & 1 & & & & & & \\ & & & 0 & & & & & \\ & & & & 0 & & & & \\ & & & & & 0 & & & \\ & & & & & & 0 & & \\ & & & & & & & 0 & \\ & & & & & & & & 0 \end{pmatrix}$$

The second linear transformation result $\mathbf{A}^2 = \mathbf{A}^1 \times \mathbf{D}^1$ is computed as:

$$\mathbf{A}^2 = \mathbf{A}^1 \times \mathbf{D}^1 = \begin{pmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

4.4 Retrieval of CAD Models based on Essential Shapes

The proposed knowledge-based multi-level simplifications enables retrieving reusable CAD models based on their essential shapes. Given a 3D query, the retrieval algorithm to be introduced in this section will compare the query with simplified shapes of archived CAD models to perform essential similarity assessment. By comparing principal shapes without unimportant details, essentially similar models can be chosen. This sort of essential shape retrieval will select CAD models that are essentially similar to the query model as the design reuse candidates, while effectively tolerating their differential modeling details.

According to comparison results, CAD models with highest similarity are retrieved and presented to designers as reusable redesign references. Because all multi-level simplification are calculated based on feature modeling precedence, designers can easily manipulate details of the retrieved parts via feature-based modifications, without violating existing feature inter-dependency. The comparison and retrieval process is defined as *essential shape matching* (ESM). The details of the ESM algorithm will be described in the following sections.

4.4.1 Generation of essential similarity descriptors

Prior to essential similarity assessment, the information of multi-level CAD model simplifications must be indexed by a concise mathematical representation for efficient comparison. The indexed information aggregates feature modeling expertise and geometric data of multi-level simplifications. Therefore, the compact mathematical representation is named *essential shape aggregation* (ESA) descriptor. An ESA descriptor captures modeling and geometric information of all simplified shapes of a CAD model. In the following paragraphs, the generation process of an ESA descriptor from a feature-based CAD model will be elaborated.

Given a CAD model, its FDAG graph can be populated using the algorithm described in section 3.3.2. Based on the normalization of the populated FDAG graph, multi-level general shape simplifications is conducted using the matrix transformation algorithm described in section 4.3. Suppose that a CAD model M has an FDAG graph G, and the corresponding multi-level simplified shapes M_i (i = 1 to n), then each simplification M_i has a corresponding FDAG sub-graph G_i. The ESA descriptor of M is generated as follows:

(1) Aggregation of modeling expertise. As knowledge reasoned results, FDAG graphs characterize feature modeling precedence expertise of CAD models. In the generation of ESA descriptor, not only feature constitutes of simplified shapes, but FDAG structures of simplifications are also included in the knowledge aggregation.

- Generate an FDAG graph of the original CAD model as G_0, in terms of the complete adjacency matrix as described in section 4.3. In G_0, feature name and geometric properties of every FDAG node are also included, along with the adjacency relations.

- Generate an FDAG sub-graph G_i (i = 1 to n) of each simplification M_i, based on the adjacency matrix transformation algorithm of section 4.3.

(2) The geometry of a simplification M_i (i = 1 to n) is characterized by SD histogram [Osada *et al.* 2002]. The rational of choosing SD histogram is that calculation of SD histogram is efficient, and this geometric characteristic shows good discrimination power for principal shapes, compared against others 3D similarity characteristics [Bespalov *et al.* 2005].

- Compute the SD histogram H_i for M_i. The computation of the D2 histogram is similar to [Osada *et al.* 2002].

- The computed histogram is a k-dimensional vector in the space $\mathbf{R}^k$, where k is the slot count of the histogram.

(3) Aggregate adjacency matrices and FDAG property graphs of the original CAD model and all simplifications into an ordered multi-dimensional representation, where the order is the sequence of multi-level simplifications (i = 0 to n, 0 means the original model) and dimension is the total count of simplification and the original model itself. The aggregation is the proposed ESA descriptor of the CAD model M.

An example visualizing the ESA descriptor generation with a mechanical part will be given in section 6.2.3.

4.4.2 Essential shape similarity

In a common mechanical design reuse scenario, if a designer wants to retrieve a particular part to reuse, he or she can sketch a 3D query that represents the basic shape of the desired model and search for all archived CAD models based on their essential similarity to the query model.

By using the ESA descriptor generation algorithm, a feature-based mechanical design can be indexed by an ESA descriptor. In this section, the generated ESA descriptor of a CAD model *M* is used for assessing the essential similarity against a user-sketched query model *Q*. The essential similarity assessment process is as follows:

(1) Calculate an SD histogram H_Q for the given 3D query model. In order to accurately characterize *Q*, it is better to use a complete shape descriptor instead of the ESA descriptor.

(2) Compare the ESA descriptor of the CAD model *M* with H_Q.

- Suppose that model *M* has *n* simplifications, its ESA descriptor will be a multi-dimensional vector aggregation, *i.e.*, $\{<H_0, G_0>, <H_1, G_1>, \ldots, <H_i, G_i>, \ldots, <H_n, G_n>\}$. Dimension 0 means the original model *M* without any simplification.

- In this research, Manhattan distance is adopted to compute the distance between H_Q and H_i (i = 0 to n), as it outperforms other metrics in the SD histogram comparison [Osada *et al.* 2002]. A k-dimensional histogram is a vector in the space $\mathbf{R}^k$, and the Manhattan distance L_1 between two vectors $X<x_1, x_2, ..., x_k>$ and $Y<y_1, y_2, ..., y_k>$ as $L_1(X,Y)=\sum_{i=1}^{k}|x_i - y_i|$.

- The essential similarity distance of the compared CAD model M to the query model Q is the shortest L_1 distance between H_Q and H_i. The corresponding M_i having the shortest distance is the representative essential shape most similar to Q. The essential similarity distance is defined as:

$$L_1^{\min}(M, Q) = \min_{i\in[0,n]} L_1(H_i, H_Q) \tag{4.8}$$

The ESM similarity of the model M to the query Q can be defined as:

$$Similarity_{ESM}(M,Q) = 1 - \frac{L_1^{\min}(M, Q)}{L_1^{\max}} \tag{4.9}$$

where $L_1^{\max}$ is the theoretical maximum Manhattan distance between D2 histograms. D2 is the function standing for the distance between any two random points on the surface of the 3D model, and D2 histogram is the probability distribution of all D2 distances. The ESM similarity is a real number. The theoretical lower bound of the similarity value of a CAD model to a query is 0, where the Manhattan distance of M to Q is equal to $L_1^{\max}$; while the upper bound is the 1 where the SD histogram of M is exactly same to the histogram of Q.

4.4.3 Essential shape matching

The above-defined ESM similarity is the rank to choose the most essentially similar shapes. ESM algorithm compares the query model with every archived CAD model by computing the essential shape similarity value to the query. According to the comparison results, the models with the highest similarity will be retrieved, as they are CAD models that are most similar to the query in terms of overall shape. The ESM algorithm is given as follows:

(1) Suppose in the design repository, there are N archived CAD models, for each of these archived model M_j (j=1 to N), we can compute its essential shape similarity as $Similarity_{ESM}(M_j, Q)$.

- A greater essential shape similarity means that the model is more similar to the query in terms of principal shape.
- If all the archived CAD models are sorted by their similarities in descending order, the first ranked model, saying M_k, is the one most similar to Q. The similarity value of M_k is formalized as:

$$\max_{j \in [1,N]} \left[Similarity_{ESM}(M_j, Q) \right] \tag{4.10}$$

(2) All essential shape matching results from the most similar to the least similar, and a percentage of top ranked models are retrieved to designers to review, and suitable retrieved results are chosen for reuse.

In the proposed ESM algorithm, not only the top-ranked model itself will be retrieved, but the attached modeling precedence knowledge will also be presented to designers as re-design reference. The modeling precedence knowledge could help designer to adjust unnecessary or differential feature details without violating any existing modeling dependency. At the same time, new features could be easily put on as pre-defined geometric constraints have been preserved during the retrieval. Therefore, the retrieved models are easy to reuse to meet new design requirements, and reused models still keep parametrically constrained. Moreover, model failure after modification issue would be greatly prevented.

Chapter 5 Retrieval Based on Partial Shape Similarity

As opposed to the essential shape retrieval, assessment on partial shape similarity matches similar portions of 3D objects. In order to assess the partial shape similarity, comparable portions should be extracted from complete CAD models. Previous 3D segmentation techniques may not perform well in mechanical part decomposition because these 3D segmentation methods only work on geometric representation. Therefore, the modeling expertise and the mechanical reusability are hardly considered in the segmentation. Without the modeling knowledge analysis, the retrieved sub-parts are mechanically meaningless patches, which are scarcely reused by mainstream CAD modeling software.

This chapter will present a CAD model retrieval method on the partial shape similarity, which includes a sub-part decomposition algorithm, the definition of the partial shape similarity, and a process to retrieve partial shapes from archived CAD models. The process of the proposing partial shape retrieval is also illustrated in Figure 5-1. The top of the figure illustrates that the knowledge for sub-part matching is extracted from the partially ordered modeling precedence of feature-based models. The knowledge is extracted from a feature directed acyclic graph (FDAG) representation, which is same to the one used in the proposed essential shape matching (ESM) method. In partial shape retrieval, a key step is to extract reusable sub-parts from complete CAD models, instead of simplifying models to essential shapes. Therefore, this chapter will apply a different analysis and interpretation on the same FDAG. As shown in right-middle of Figure 5-1, in order to identify reusable and decomposable sub-parts, a vertical FDAG partitioning algorithm will be proposed to identify a reasonable sub-part. These identified sub-graphs can be extracted from completed models as reusable sub-parts. As shown in the bottom of the figure, once all sub-parts are extracted, these sub-parts will be compared with a query model in terms of partial shape similarity, and the most similar ones to the query will be presented to designers for further reuse. As the sub-part decomposition is conforming to the existing modeling dependency expertise embedded in original CAD models, the reuse of these sub-parts will be fairly easy.

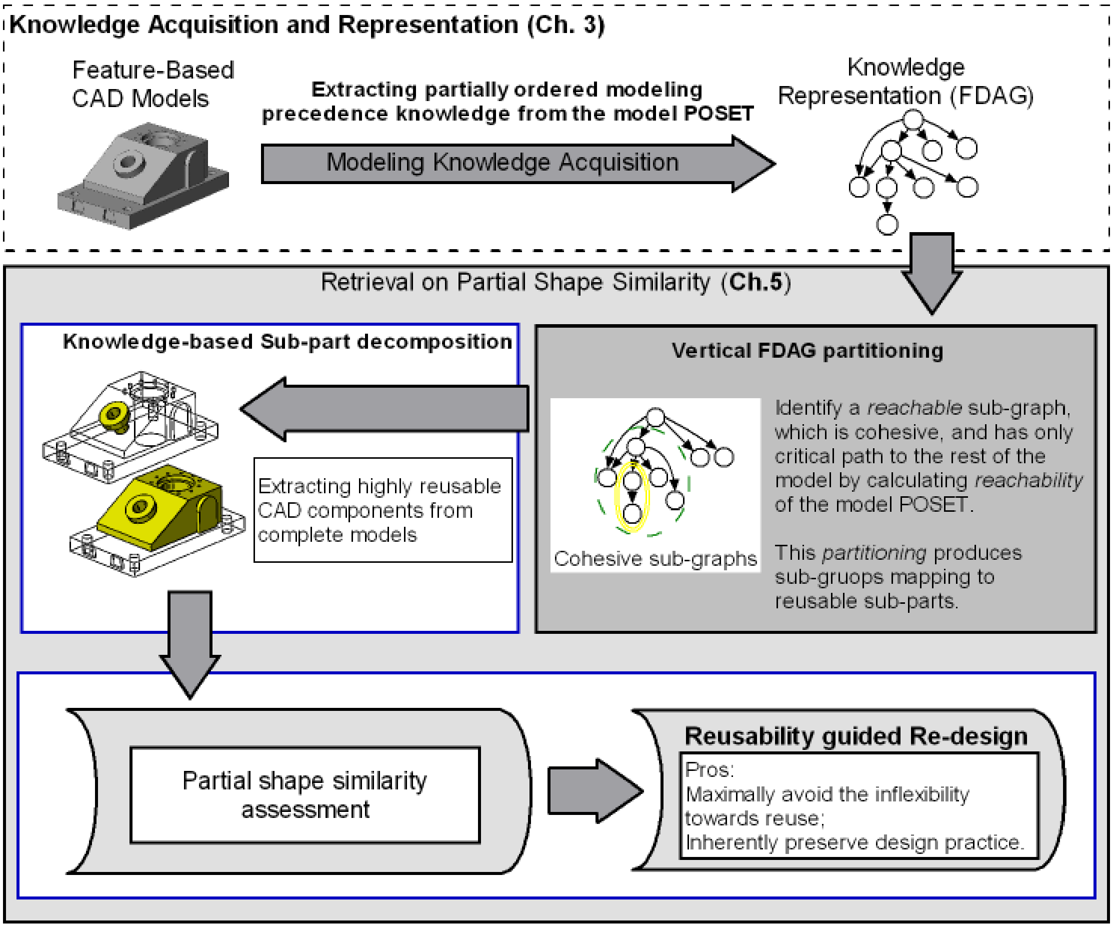


Figure 5-1. Flow chart of the partial shape retrieval method

5.1 Partial Shape Retrieval

The retrieval on partial shape similarity requires a similarity assessment algorithm that can find out sub-part correspondence within a complete mechanical part. In addition to the retrieval, the found sub-part should be easily decomposed from the original complete model, and can be conveniently integrated into another mechanical part for reuse.

Current partial shape matching algorithms do not work well for partial retrieval and reuse of mechanical CAD models. Typical 3D shape segmentation techniques, including fuzzy clustering, concavity detection, or topological critical point based, are all highly dependent on the geometric factors of compared models. Therefore, in a geometric segmentation algorithm, a small change in geometry will result in a large difference in segmentation results. Such geometry-sensitive segmentation might prevent subsequent sub-part matching algorithm from retrieving satisfactory results. Furthermore, after sub-part matching retrieves some partial shapes from complete models, the reuse of these retrieved results will become another difficult problem because merging two arbitrary freeform surfaces, *i.e.*, stitching is still a challenging research topic.

In order to address the aforementioned obstacles and retrieve reusable CAD partial models, a sub-part decomposition algorithm is proposed to extract reusable mechanical sub-parts from complex CAD models, by applying vertical partitioning on the FDAG representation. The designed vertical partitioning is to find out reachable sub-graphs, which are internally cohesive and externally decomposable from the rest of the FDAG representation. All decomposed sub-parts are indexed into a concise representation, which is named the *partial shape aggregation* (PSA) descriptor. With the help of the ESA descriptor, partial similarity assessment can be done to search partial CAD models, using a sketched sub-part as the query.

The knowledge-based partial shape matching algorithm is reuse-oriented as original design intents and expertise have been utilized for the proposed sub-part retrieval and migration; therefore, reuse inflexibility is avoided. In addition to the reusability, design knowledge is inherently preserved and transferred to the new design, which not only realizes functional shape reuse, but also maximally promotes the reuse of design knowledge.

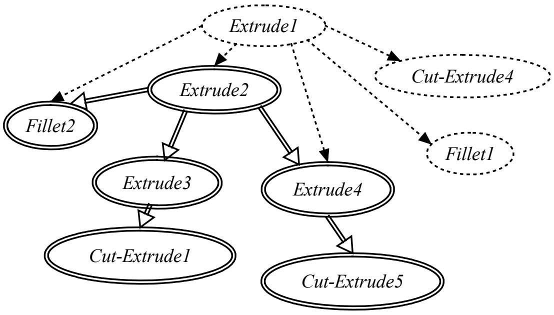
5.2 Knowledge-Based Vertical Partitioning

The vertical partitioning algorithm developed for partial shape retrieval is based on the feature modeling precedence presented in Chapter 3. The modeling precedence of a CAD model is captured by the FDAG graph representation.

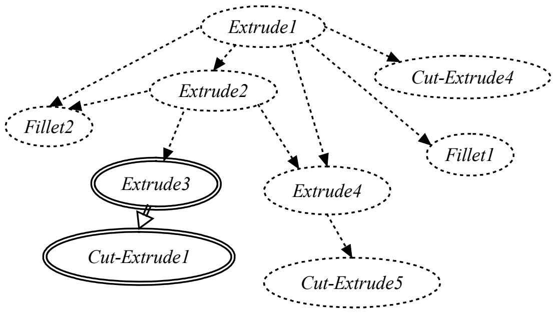
Let us revisit the bracket shown in Figure 4-7a. The inter-dependency relations of all features can be illustrated by an FDAG graph shown in Figure 4-7b. In the figure, a directed edge between two nodes indicates that the source node can reach another sink node. In the graph theory, the reachability is a property, which starts from one vertex in a directed graph to travel to another vertex. For a directed graph $D = (V, E)$, the reachability of the vertices is the transitive closure of its edge set E. That is to say that for any transitively reachable vertex pair (x, y), there exist a chain of vertices $x = v_0, v_1, \ldots, v_n = y$ where all (v_{i-1}, v_i) are also in E for all $1 \leq i \leq n$.

Since FDAG is a directed acyclic graph, the reachability in an FDAG is also a partially ordered relation on graph vertices. The partially ordered reachability is transitive; hence the transition closure (TC) can be computed on each node. For example, the reachability TC of node *Extrude2* is the sub-graph {*Extrude2*, *Fillet2*, *Extrude3*, *Cut-Extrude1*, *Extrude4*, *Cut-Extrude5*}, in which three reachable chains are included. The transition closure sub graph of *Extrude2* is shown in Figure 5-2a.

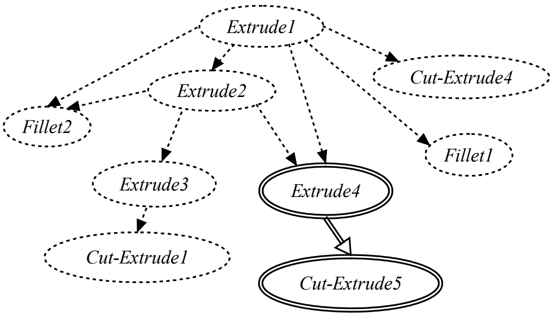
Similarly, the transition closures of the FDAG nodes *Extrude3* and *Extrude4* are {*Extrude3*, *Cut-Extrude1*} and {*Extrude4*, *Cut-Extrude5*}, which are also shown in Figure 5-2b-c.



(a) Transition closure of *Extrude2*



(b) Transition closure of *Extrude3*



(c) Transition closure of *Extrude4*

Figure 5-2. The transition closure sub-graph (shown in double-lines) of the FDAG in Figure 4-7b

The nodes in an FDAG are partially ordered, which means that only a pair of nodes having geometric constraint relations is ordered. In other words, some nodes without modeling relationship can be considered as unrelated. This means that within the same FDAG graph, there can be two nodes which are not reachable to each other. Also, multiple FDAG sub-graphs can be disjointed in terms of reachability. For example, in Figure 5-2(b-c), the figures shows transition closures of *Extrude3* and *Extrude4*, respectively. At the same time, these two sub-graphs correspond to geometrically cohesive sub-parts, which are also functionally complete from a mechanical perspective. As illustrated in Figure 5-3, transition closures of *Extrude3* and *Extrude4* represent the shaft housing and rib sub-parts, respectively. Using the reachability-based modeling independence deduced, the vertical partitioning of FDAG provides a way to segment CAD sub-parts.

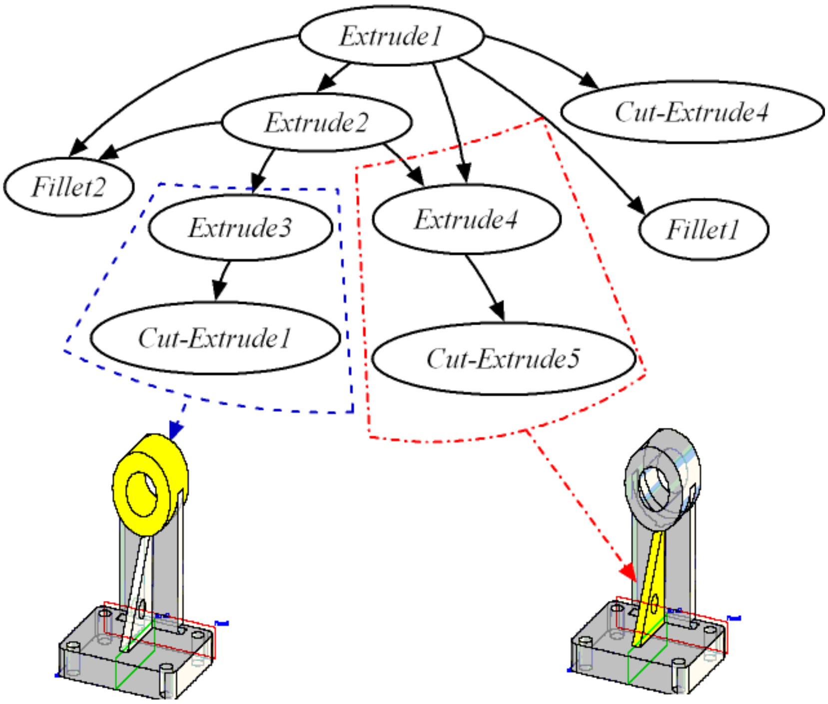


Figure 5-3. Reachability-based vertical partitioning on the normalized FDAG graph and their geometry correspondences

Unlike the horizontal partitioning proposed in Chapter 4.2 that simplifies FDAG hierarchies from minimal elements to maximal elements, the reachability-based vertical partitioning can be adopted to decompose partial shapes from complete CAD models. The next section will introduce sub-part decomposition of CAD models based on the proposed FDAG vertical partitioning, including an algorithm of validation on reusable partial matching candidates.

5.3 Sub-Part Decomposition of CAD Models

A logical sub-part decomposition of a CAD model should identify decomposable CAD model portions which have reasonable geometry meanings. Furthermore, it will be better if the decomposed sub-part is easy to be relocated to another 3D design.

In the previous section, the vertical FDAG partitioning is proposed to extract CAD sub-parts using the reachability of modeling dependencies. Using the reachability-based partitioning, a complicated CAD model can be decomposed into sub-parts. Taking a pusher pad part as example, its feature model is shown in Figure 5-4. The FDAG of the pusher pad is illustrated in Figure 5-5a. As the reachability is a binary relation, all the reachability relationships can be captured by a binary matrix, where each matrix entry is either zero or one. The reachability is transitive because it is also a partially ordered relationship. Therefore, in Figure 5-5b, the transitively reachable relationships are computed and superimposed as double-lines, on top of Figure 5-5a.

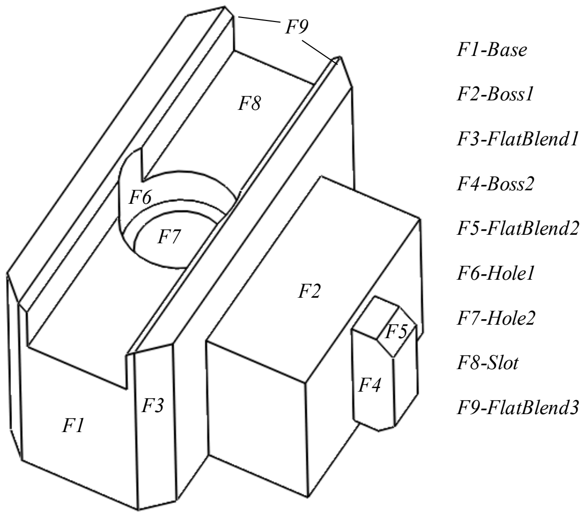
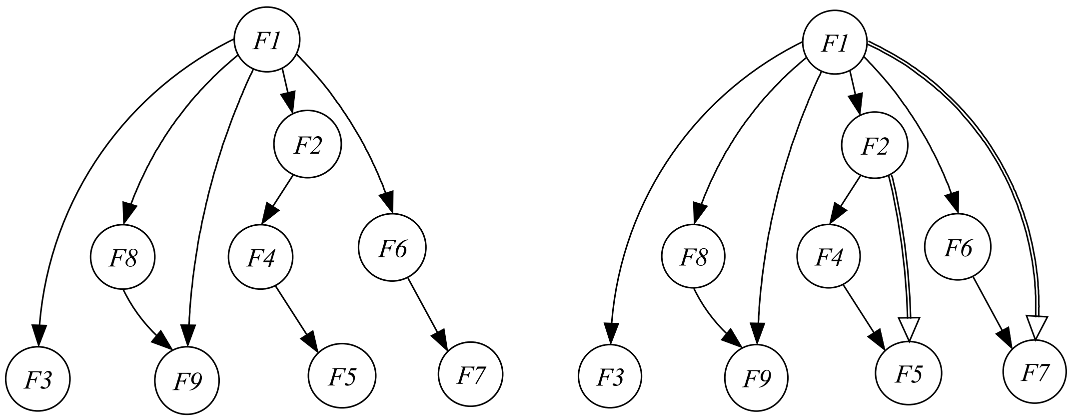


Figure 5-4. A pusher pad and its feature model



(a) FDAG graph of the pusher pad (b) Highlighted transitive reachability

Figure 5-5. The FDAG graph of the pusher pad shown in Figure 5-4

Using the superimposed FDAG shown in Figure 5-5b, every FDAG sub-graph which is corresponding to a tentative CAD sub-part can be easily decomposed by applying the vertical FDAG graph partitioning. Taking vertex *F2* as the example, its vertically partitioned sub-graph will be a vertex set, in which every node has a direct or transitive reachability from node *F2*. Especially, *F2* itself is also included in the sub-graph as the root node because the partially ordered reachability is also reflective.

All vertical partitioned sub-graphs are listed in Table 5-1, and in the table, SG_3 is the sub-graph whose root is *F2*.

Table 5-1. Sub-graphs elements partitioned by the vertical FDAG partitioning

Sub-Graph	**Sub-Graph Root**	**Sub-Graph Elements**
SG_1	*F1*	*F1*, *F3*, *F8*, *F9*, *F2*, *F4*, *F5*, *F6*, *F7*
SG_2	*F2*	*F2*, *F4*, *F5*
SG_3	*F8*	*F8*, *F9*
SG_4	*F4*	*F4*, *F5*
SG_5	*F6*	*F6*, *F7*

Every FDAG sub-graph corresponding to the Table 5-1 is illustrated in Figure 5-6.

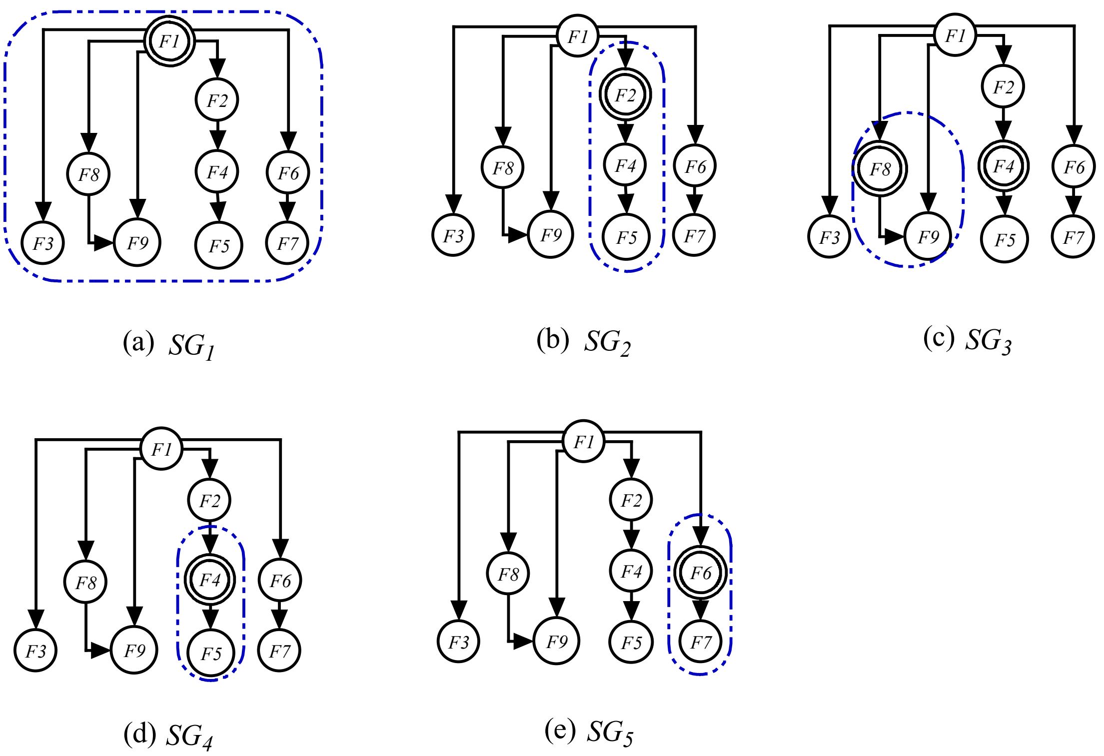


Figure 5-6. FDAG sub-graphs partitioned by the vertical FDAG partitioning

However, not every sub-graphs partitioned by the FDAG reachability is a reusable sub-part from the viewpoint of mechanical reuse. Ideally, a reusable sub-part can be easily decomposed from the original CAD model, and can also be conveniently migrated to another target model. Moreover, the merged model ought to be not only geometrically watertight, but also well-constrained in terms of feature-based modeling. This requires that a reachability-partitioned sub-graph should meet certain domain knowledge related criteria so that it would be a reasonable sub-part candidate for mechanical partial shape retrieval. The problem with current feature modeled CAD parts is that a single design is composed of many features which have inter-dependencies. Although these dependencies are powerful during the design phase, it becomes constraining for design reuse. Over complicated inter-dependencies between partitioned sub-parts and the rest of a complete CAD model would become an obvious obstacle for mechanical reuse. Therefore, a reusable sub-part partition ideally has limited external dependencies to the outside. This ensures that the subpart is independent from the rest of the model except having limited dependencies.

The key to decompose a reusable sub-part with limited external dependencies is to find out *articulation points* of the FDAG graph. In graph theory, a vertex v of a connected graph G is an articulation point (AP) if the deletion of v produces a graph that has more connected components than the original G. In Figure 5-5, the nodes *F1*, *F2*, *F4* and *F6* are APs of the FDAG graph. In an FDAG graph, if an AP can be identified and the edges directed can be removed, a sub-graph sourcing from the AP will be separated from the complete FDAG as the only connection via the AP is broken. Correspondingly, a tentative sub-part is segmented from the CAD model.

The tentative sub-part partitions can be evaluated using the aforementioned mechanical criteria. First of all, the root of the FDAG cannot be the AP, as shown in

Figure 5-6a, otherwise the complete part would be its accessible sub-graph. Therefore, SG_1 is not a valid sub-part partition. On examining other partitioning among Figure 5-6(b-e) by deleting all edges incident to each sub-graph root and the root itself, node *F8* is identified as an invalid AP. Therefore, SG_3 is not a valid sub-part.

The aforementioned criteria will be used to evaluate the reusability of sub-parts decomposed from vertical partitioning. As a valid sub-part is enforced to be geometrically connected, it will be more straightforward to designers, to understand modeling inter-dependencies and original design intents, hence facilitating the work-in-process. Furthermore, a validated sub-part only has limited external dependencies, which makes it easy to decouple limited dependencies from the rest of the model, and relocate the sub-part to a new CAD model by resolving few dependencies. An example visualizing the migration of a matched sub-part will be illustrated in 6.3.2.

The sub-part segmentation algorithm consisting of the vertical FDAG partitioning and meaningful sub-part validation is given as follows:

(1) Identify APs $\{v_1, \ldots, v_i, \ldots, v_n\}$ of the FDAG G of a CAD model M. The variable i indicates the sequence of sub-graphs, ranging from 1 to the number of sub-graphs n. Traverse every identified AP:

- For an AP v_i visited, compute the transitive closure of v_i as its tentative sub-graph.

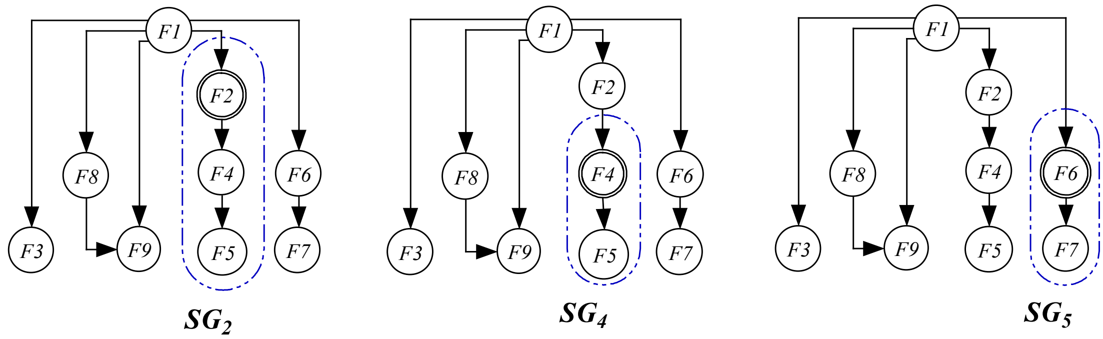
(2) Validate all tentative sub-graphs $\{G_1, \ldots, G_i, \ldots, G_n\}$, which are partitioned by APs $\{v_1, \ldots, v_i, \ldots, v_n\}$, respectively. The sub-graphs $\{G_1, \ldots, G_i, \ldots, G_n\}$

correspond to sub-parts $\{P_1, \ldots, P_i, \ldots, P_n\}$. A valid sub-graph represents a sub-part that is easy to extract and reuse.

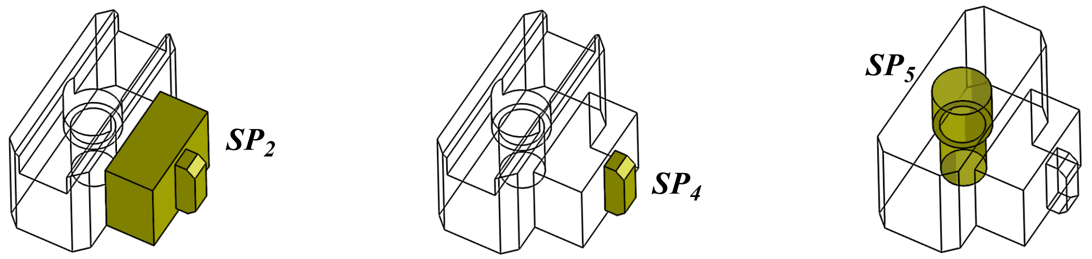
- If a tentative sub-graph G_i equals to the FDAG G, it is not valid because the corresponding sub-part P_i is the complete model M;
- If G_i introduces more incoming edges than those pointing to v_i, the sub-graph is not valid. The rationale is that more edges from the external means that additional constraints need to be resolved for relocating the partitioned sub-part. Assume $D^-(\{v_i\})$ and $D^-(G_i)$ are the set of external dependencies of AP v_i and sub-graph G_i, respectively, the following must hold:

$$D^-(G_i) \subseteq D^-(\{v_i\}) \tag{5.1}$$

Applying the proposed sub-part decomposition algorithm on the part in Figure 5-4, a total of 5 tentative sub-graphs are segmented, as listed in Table 5-1. After validating these tentative sub-graphs, SG_1 and SG_3 are identified as invalid sub-part segmentation, because of their inability to reuse. Three valid sub-graphs, SG_2, SG_4 and SG_5 are listed in Figure 5-7a. These sub-parts are geometrically connected CAD components that have limited dependencies to the rest the model so that the desirable reuse properties, such as cohesiveness and decoupling are satisfied. Furthermore, the knowledge-based sub-part segmentation is not a simple geometrical division on CAD model shape. It can be clearly observed that SP_4 is hierarchically contained in SP_2, which cannot be easily obtained by geometric-based segmentation algorithms. Especially, SP_5 shows a counter-bore sub-part only consisting of negative features (*F8* and *F9* are suppressed to show the complete counter-bore shape). Such subtractive component is hard to be extracted by purely geometrical segmentation algorithms.



(a) Valid sub-graphs

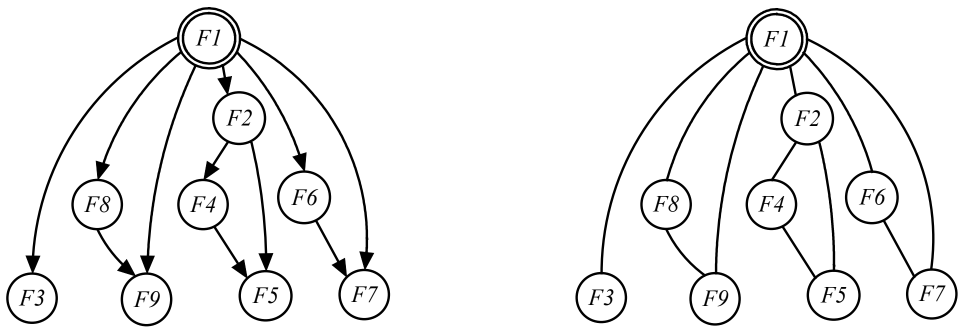


(b) Reusable sub-parts

Figure 5-7. The valid sub-graph segmentations and the corresponding sub-parts

The knowledge-based sub-part segmentation method described above has been implemented by an efficient matrix computation algorithm. In the development of the algorithm, a key point is to identify articulation points (AP) of the knowledge-based FDAG representation. A depth-first search based algorithm was proposed by Tarjan [1972] to determine strongly connected sub-graphs of directed graphs. Here the AP is the entry vertex of the depth-first search. However, the strongly connected property only exists in directed cycle graphs. For example, let G be a directed graph, and G is said to be strongly connected if and only if for each pair of vertices, v, w in G, $v \neq w$, there are paths $v \rightarrow w$, and $w \rightarrow v$. Therefore, a strongly connected component contains at least one cycle. Obviously, this method is not suitable for FDAG representation, which is an acyclic directed graph. In this research, an AP identification algorithm is proposed for acyclic FDAG graphs by determining APs of the un-directed approximation of FDAG representation. The rational is that the node adjacency in an un-directed approximated graph is symmetrical and cyclic, which is stronger than FDAG representation that is asymmetrical and acyclic. Therefore, once a node is identified as an AP on the equivalent un-directed approximation, the deletion of this AP vertex from the FDAG will also leave the FDAG disjointed. The proposed AP identification algorithm is based on the work presented by Hopcroft and Tarjan [1973], who developed an linear-time algorithm for finding out bi-connected sub-graphs of an un-directed graph. Un-directed graphs are equivalent to bi-directed graphs if no negative edges are investigated [Schrijver 2003].

An un-directed approximation of FDAG representation can be obtained by inserting missing reverse edges. Taking Figure 5-8a as an example, by inserting missing reverse edges, the corresponding un-directed approximation (equivalent to bi-directed graph with directed edges combined) is shown in Figure 5-8b.



(a) Directed FDAG graph (b) Un-directed approximation

Figure 5-8. An FDAG graph of the pusher pad and its un-directed approximation

The AP identification algorithm is presented as follows. Let an FDAG graph be $G = \{V, E\}$, then $H = \{V, E_H\}$ is the un-directed graph approximation of G. If in the context of $G, \forall(V,W) \in E$, there will be an undirected edges (V,W) and $(W,V) \in E_H$ in the un-directed approximation, which are the union of forward and reverse edges of the directed graph instance. Let $H_\delta = \{V, E_\delta\}$ be the depth-first spanning tree of the graph approximation $H = \{V, E_H\}$, and a node $a \in V$ will be an AP of H if and only if:

i. Node a is the root of H_δ , and a has at least two children in H_δ, or

ii. Node a is a non-root element of H_δ, and there exist vertices v, w $\in V$ such that a directed edge $(a,v) \in E_\delta$ and $LOW(v) \geq DFN(a)$, and v is not an ancestor of w in H_δ

The proof can be found in [Cormen *et al.* 2001]. In the definition, $DFN(a)$ is the depth first number of traversing the depth-first spanning tree H_δ; while $LOW(v)$ is the smallest depth first number which can be reached from v using a directed path containing descendents of v and at most one back edge.

Let the graph approximation $H = \{V, E_H\}$ have a depth-first spanning tree $H_\delta = \{V, E_\delta\}$ with back edge set B, the value $LOW(x)$ where $x \in V$ is defined as:

$$LOW(x) = \min(\{DFN(x)\} \cup \{LOW(y) \mid (x,y) \in E\delta\} \cup \{DFN(y) \mid (x,y) \in B\} \qquad (5.2)$$

First of all, an ordering on all graph nodes of Figure 5-8b is defined according to the DFS traversal sequence, which is given as:

$Seq^{DFS}(H=(V, E_H)) : F1,\ \ F3,\ \ F8,\ \ F9,\ \ F2,\ \ F4,\ \ F5,\ \ F6,\ \ F7$

Secondly, according to these definitions, the values of $DFN\ (V_i)$ and $LOW\ (V_i)$ for a node V_i ($1 \le i \le |V|$) are given as:

$$DFN\ [\ 1{:}9\ ] = \{\ 1, 2, 3, 4, 5, 6, 7, 8, 9\ \}$$

$$LOW\ [\ 1{:}9\ ] = \{\ 1, 2, 1, 1, 5, 6, 7, 8, 9\ \}$$

The APs of the approximated graph shown in Figure 5-8b can be identified as *F1*, *F2*, *F4*, and *F6* using the above AP definition. It is easy to verify that the identified APs are also cut vertices that can split the FDAG graph into separated.

Once APs of an FDAG graph are identified, reachable sub-graphs rooting from these APs can be computed by matrix calculations. The matrix representation is built to represent the reachable relationship among FDAG nodes. Analogous to the adjacency matrix defined in section 4.3, we define the reachability matrix **R** $= [r_{ij}]_{n \times n}$ in which the rows correspond to source graph nodes of the reachability and the columns are sink nodes. In this reachability matrix **R**, every element r_{ij} has a Boolean value indicating the reachability between from a source node v_i to a sink v_j:

$$r_{ij} = \begin{cases} 1, & v_i \text{ is reachable to } v_j \\ 0, & \text{othewise.} \end{cases} \tag{5.3}$$

Although the definition of reachability matrix is similar to Eq. (4.3), these two matrixes have different meanings and varying properties. For Eq. (5.3), it describes the binary reachability (or the accessibility) relationship of the FDAG representation. The adjacency matrix given in Eq. (4.3) corresponds to a directed acyclic graph, which corresponds to a strictly partial order set (*i.e.*, ir-reflexitive, asymmetric and transitive). However, from the viewpoint of the reachability matrix given in Eq. (5.3), the reachability is only a partial order relationship, which means that an FDAG node can be considered accessible to itself.

The sequence of row and column of the proposed reachability matrix is the same as the DFS traversal ordering of FDAG nodes. The reachability matrix is shown in Table 5-2. It can be seen that all diagonal elements of the matrix are 1 due to the fact that the reachability is partially ordered, instead of strictly partially ordered.

Table 5-2. The reachability matrix of the pusher pad

v_j		*j=1*	*j=2*	*j=3*	*j=4*	*j=5*	*j=6*	*j=7*	*j=8*	*j=9*
v_i		*F1*	*F3*	*F8*	*F9*	*F2*	*F4*	*F5*	*F6*	*F7*
i=1	*F1*	**1**	**1**	**1**	**1**	**1**	0	0	**1**	0
i=2	*F3*	0	**1**	0	0	0	0	0	0	0
i=3	*F8*	0	0	**1**	**1**	0	0	0	0	0
i=4	*F9*	0	0	0	**1**	0	0	0	0	0
i=5	*F2*	0	0	0	0	**1**	**1**	0	0	0
i=6	*F4*	0	0	0	0	0	**1**	**1**	0	0
i=7	*F5*	0	0	0	0	0	0	**1**	0	0
i=8	*F6*	0	0	0	0	0	0	0	**1**	**1**
i=9	*F7*	0	0	0	0	0	0	0	0	**1**

Using the reachability matrix given, the reachable sub-graphs of each FDAG APs can be obtained by computing the reachability transitive closure. In a directed graph $G = \{V, E\}$, the successor set of a vertex $v \in E$, denoted by *Succ*(v), is the set of vertices that are directly reachable from v. The vertex set adjacent to v in the transitive closure is the same as the *Succ*(v) in the original graph G. Therefore, computing the transitive closure is equivalent to recursively computing the successor set for every vertex in G. The recursive computation procedure can be constructed intuitively step by step. The first step of transitive closure is defined as:

$$G^0 = G \tag{5.4}$$

and the reachability relation between a and b is defined as $\overrightarrow{ab}$, the following steps can be given as:

$$G^i = G^{i\text{-}1} \cup \left\{ \overrightarrow{ab} \;\middle|\; \exists b \text{ where } \overrightarrow{ab} \in G^{i-1} \text{ and } \overrightarrow{bc} \in G^{i-1} \right\} \tag{5.5}$$

Each graph in the recursive procedure takes the reachability relation from the previous graph, and adds new reachability to make the final graph more transitively reachable. The Floyd-Warshall Algorithm [Floyd 1962, Warshall 1962] is a graph analysis algorithm for finding the shortest paths. This algorithm can be used to solve transitive closure (TC) problem as well as once a path is found between vertices u and v, the transitive reachability between them is also determined. Suppose $|V|$ is the number of vertices in a graph, let an FDAG be G=(V, E), and take the reachability matrix $\mathbf{R}_{|V|\times|V|}$ on the G as input, the Floyd-Warshall Algorithm of TC computation is given as:

```
Start with R*=R, for each k from 1 to |V|
  For each i from 1 to |V|
    For each j from 1 to |V|
      If R*(i,k)=1 and R*(k,j)=1
      Then compute AND value of row i and j, and replace row i by it.
    Go on to the next j
  Go on to the next i
Once processed each i, go on to the next k.
```

Using the above algorithm, the computation complexity will be O ($|V|^3$); while the space complexity is O ($|V|^2$) because the transitive reachability can be directly updated on the original matrix. The reachability matrix **R** is given as:

$$\mathbf{R} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

By using the Floyd-Warshall Algorithm, the transitive closure **R*** can be computed as:

$$\mathbf{R^*} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

It can be conventionally observed in **R*** that, all diagonal elements are 1 because of the reflexive reachability; and each row represents the transitively reachable relations from source nodes to sink nodes. Taking the row 5 for example, it represents all reachability relationships from element *F2* (i=5). Three reachable nodes {*F2*, *F4*, *F5*} is the closure of node *F2* in terms of the reachability.

In this research, the flow graph is chosen to be the data structure of a reachable sub-graph led by an AP node. A flow graph can be defined as $F(s) = (V, E, s)$, where F a directed graph with a starting vertex (dominator) $s \in V$ such that every vertex in V is reachable from s. Taking the reachability closure of *F2* as an example, the dominator vertex is *F2*, and the reachable sub-graph is represented by *SG*(*F2*) = ({*F2*, *F4*, *F5*}). All reachable sub-graphs shown in **R*** can be summarized in Table 5-3.

Table 5-3. Reachable sub-graphs shown in **R***

Sub-Graph Dominator Nodes	**Reachable Sub-Graph Elements**
F1	*F1*, *F3*, *F8*, *F9*, *F2*, *F4*, *F5*, *F6*, *F7*
F3	*F3*
F8	*F8*, *F9*
F9	*F9*
F2	*F2*, *F4*, *F5*
F4	*F4*, *F5*
F5	*F5*
F6	*F6*, *F7*
F7	*F7*

However, among these dominator vertices, only *F1*, *F2*, *F4*, and *F6* are selected as APs of the FDAG. With further evaluation on the selected sub-graphs dominated by corresponding APs, it is easy to find that the reachable sub-graph dominated by *F1* is the complete FDAG itself. Therefore, this sub-graph fails to meet the reusability criteria discussed in section 5.3. The final sub-graph segmentations evaluated are {*F2*, *F4*, *F5*}, {*F4*, *F5*}, and {*F6*, *F7*}. These results are consistent to the valid segmentations shown in Figure 5-7.

5.4 Retrieval of CAD Models based on Partial Shapes

The proposed sub-part decomposition can enable CAD component retrieval based on their partial shapes. Given a 3D query of a sketched partial shape, the retrieval algorithm to be introduced in this section will match segmented sub-parts to perform partial similarity assessment.

The proposed partial shape matching algorithm selects CAD sub-parts similar to the query model. According to the selected partial matching results, sub-parts with the highest similarity are presented to designers as reusable candidates. A retrieved sub-part can be conveniently decomposed from limited external dependency and relocated to another mechanical part as the matched CAD components are extracted based on the reachability of existing feature inter-dependencies. Most importantly, all modeling constraints within the relocated part will be preserved during the reuse process, therefore facilitating transfer of design knowledge and promoting further design reuse.

Essentially, there are three steps in retrieving feature-based CAD models based on their partial shape similarity: extraction of sub-parts, generations of partial shape descriptors, and retrieval of partial shapes. In the following sections, these steps are described.

5.4.1 Generation of partial similarity descriptors

Prior to partial similarity assessment, the information of segmented sub-part must be indexed by a concise mathematical representation. The indexed information characterizes not only geometric data of sub-part segmentation, but also captures the feature modeling dependencies of the sub-part. The concise representation is named as *partial shape aggregation* (PSA) descriptor. In the following paragraphs, the generation process of a PSA descriptor of an extracted sub-part will be elaborated.

In this research, each feature-based CAD part has a corresponding FDAG graph representation for reuse-oriented retrieval purpose. By applying FDAG vertical partitioning on an FDAG graph, decomposable sub-graphs can be identified and every sub-graph corresponds to a cohesive and decoupling sub-part. Once a sub-part is extracted from the complete CAD model as a candidate partial shape, a PSA descriptor will be generated for this sub-part. The generation process of the PSA descriptor is given as follows:

(1) Calculation of modeling and geometry information of the sub-part. The PSA descriptor not only indexes the geometry of the sub-part, but also captures the embedded modeling characteristics of the sub-part.

- Compute the geometric characterization of a sub-part. The geometry of the sub-part is characterized by the SD histogram. Suppose that a CAD model M has an extracted sub-part, says P, the D2 histogram H_P is the geometric characteristic of P. The computed SD histogram is a k-dimensional vector in the space $\mathbf{R}^k$, where k is the slot count of the histogram.

- Characterize the internal modeling dependencies of the sub-part. The internal modeling dependencies are characterized by a flow graph $F(s) = (V, E, s)$, where node s is the dominator of the graph, which is equivalently the AP node of the extracted sub-part.

- Index the external dependencies between the sub-parts and the rest of the complete CAD model. External dependencies of the sub-part are indexed by the FDAG graph of the CAD model, and the external dependences are the adjacencies incident to the AP node.

(2) Aggregating modeling and geometry information into the final PSA descriptor.

5.4.2 Partial shape similarity

By applying the vertical partitioning on the FDAG, cohesive and decoupling sub-parts can be extracted from complete CAD models. These extracted sub-parts are reusable partial shape candidates, which are characterized by PSA descriptors. In this section, all extracted sub-parts $\{P_1, P_2, \ldots, P_n\}$ from various complete CAD models are compared against a user-sketched query Q to find similar partial shapes. The partial shape similarity is assessed as follows:

(1) Calculate the SD histogram H_Q for the given 3D query model Q.

(2) Compare the PSA descriptor of sub-parts P_i with H_Q (i ranges from 1 to n).

- Compute the Manhattan distance between H_Q and the histogram H_{Pi} of P_i, $L_1(H_{Pi}, H_Q)$, which represents the dissimilarity from P_i to the query Q in terms of geometry.
- The *partial shape similarity* of P_i to Q is defined as:

$$Similarity_{PSM}(P_i, Q) = 1 - \frac{L_1(H_{Pi}, H_Q)}{L_1^{\max}} \tag{5.6}$$

where value $L_1^{\max}$ is the upper limit of SD D2 histogram dissimilarity. The PSM similarity is a real number. The theoretical lower bound of the similarity is 0, where the Manhattan distance of P_i to Q is $L_1^{\max}$; while the upper bound is 1, which means that the SD histogram of P_i is exactly same to the histogram of Q.

5.4.3 Partial shape matching

The proposed partial shape similarity is adopted as the criterion to sort candidates in the *partial shape matching* (PSM) algorithm. PSM evaluates every extracted sub-part by computing their partial shape similarity to the one of the query model. Based on the outcome of the evaluation, sub-parts with the highest similarity will be retrieved. The PSM algorithm is described as follows:

(1) In the design repository, there are total M sub-parts extracted from all archived CAD models. These sub-parts are labeled as P_i (i=1 to M).

(2) The partial shape similarity of P_i (i=1 to M) to the query Q can be computed according to the algorithm proposed in section 5.4.2.

- A greater partial shape similarity means that compared sub-part is more similar to the query, compared against those having less similarity.

- If all the extracted CAD partial components are sorted by their partial shape similarities in descending order, the first ranked model, saying P_k, is the one most similar to Q. The similarity value of most similar model P_k is formalized as:

$$\max_{k \in [1,M]} \left[Similarity_{PSM}(P_k, Q) \right] \tag{5.7}$$

(3) Matched results from most similar to least similar will be retrieved to designers.

In the proposed PSM algorithm, not only the top-ranked partial shapes are presented to designers for reviewing its geometrical reusability, but also their modeling expertise embedded in the PSA descriptor will be used as re-design references to evaluate its mechanical reusability. First of all, the modeling dependency between a sub-part and its host CAD model will help designers smoothly externalize the sub-part as a reusable component. Secondly, in the reuse of this sub-part, the same external dependency knowledge will facilitate to find out a possible solution to re-locate the retrieved sub-part to a new CAD model. Finally, the internal modeling dependencies of the sub-part, another piece of domain knowledge embedded in the PSA descriptor, is serving as the media of knowledge transfer during the reuse, to preserve the original design intent of the reused CAD sub-part. Therefore, the reused sub-part is still fully constrained and well integrated with the target CAD model in a feature modeling manner, which makes future retrieving of the reused sub-part feasible and straightforward.

Chapter 6 Results and Discussion

In this chapter, the reuse-oriented retrieval algorithms described in this research are implemented and evaluated using real-world CAD models. The evaluations are conducted using standard precision-recall curves to verify the algorithmic effectiveness, and experimental results are discussed as well. To build a realistic-scale evaluation dataset, over six hundred CAD models are collected from online part repositories and classified by experienced designers.

Furthermore, a prototype system has been built upon a commercial modeling application. In the built prototype system, the power of the essential shape matching (ESM) and partial shape matching (PSM) algorithms is leveraged to effectively bridge the identified gap between current CAD model retrieval and reuse. Several case studies are discussed to prove that the proposed approaches are feasible to perform the reuse-oriented retrieval under the industrial settings.

6.1 System Implementation

A prototype system has been implemented to verify the effectiveness of the proposed algorithms. In the system implementation, it is vital to show that how the proposed approaches are facilitating 3D CAD model retrieval as well as downstream reuse activities, particularly under current industrial settings. In the following sub-sections, a use case diagram is shown to capture system-level requirements of reuse-oriented retrieval activities, and the architecture of the prototype system is presented.

6.1.1 Requirements of reuse-oriented retrieval

Nowadays, mainstream mechanical design is still dominated by 3D parametric modeling systems. Designers generally need a 3D parametric modeling system seamlessly supporting retrieval functionality. Within the same modeling interface, the system is able to cover a complete process of model querying, retrieval, redesign analysis, and reuse realization. These requirements are consolidated into a use case diagram, which is illustrated in Figure 6-1.

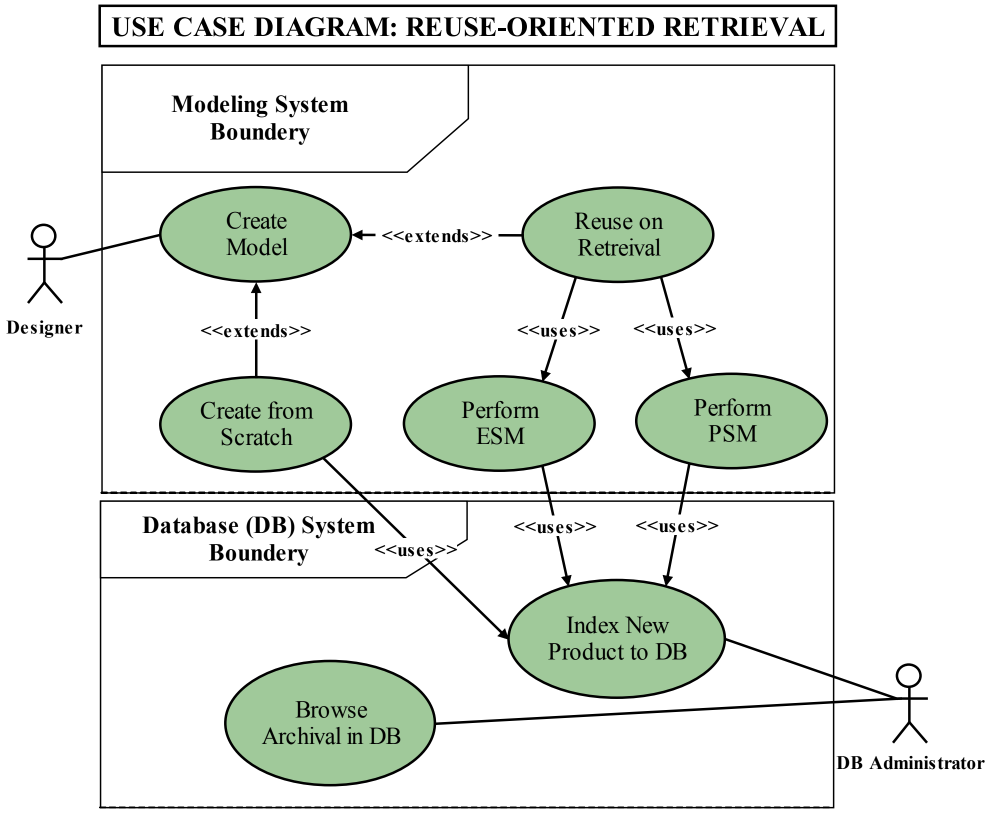


Figure 6-1. Use case diagram of reuse-oriented retrieval activities.

In this use case diagram, the upper frame shows the system boundary of a CAD modeling interface, in which users can either create a new design from scratch or retrieve an existing design to reuse. Also in the same interface, the ESM and PSM algorithms have been implemented as an extension and integrated into the modeling system. Therefore, the modeling system is able to support real-time sketch and search, enabled by the seamless integrated extension. Users can build a basic shape of a desired part; in the meanwhile, the implemented extension will compare the inputted query with archived models using ESM and PSM algorithms, and present the most similar ones to designers. This *sketch and search* mechanism maximizes reuse possibility as reusable candidates are proactively presented to users in a synchronous manner while they are designing, instead of opening a new window to search, which could be an obstacle to break design and reuse.

No matter how a design is created, either building from scratch or re-designing an existing design, the newly created will be indexed again into design database, where all archived realistic-scale CAD models have been indexed by ESA / PSA descriptors, as proposed in sections 4.4.1 and 5.4.1, respectively. In this way, new designs will be characterized for future retrieval and reuse, and database consistency is also well-maintained.

6.1.2 Implementation of the prototype system

All these requirements have been implemented into the prototype system, and the high-level architecture of the prototype is illustrated in Figure 6-2. The CAD modeling function is provided by a traditional 3D modeler, which also serves as a platform to enable 3D query and variational reuse. The modeler is shown in the center of the figure. The outer tier is the reuse-oriented retrieval extension, which encapsulates knowledge-based CAD model retrieval functionalities based on a 3D query input. Once a model is located by the retrieval extension, it will be presented to designers as a candidate redesign reference.

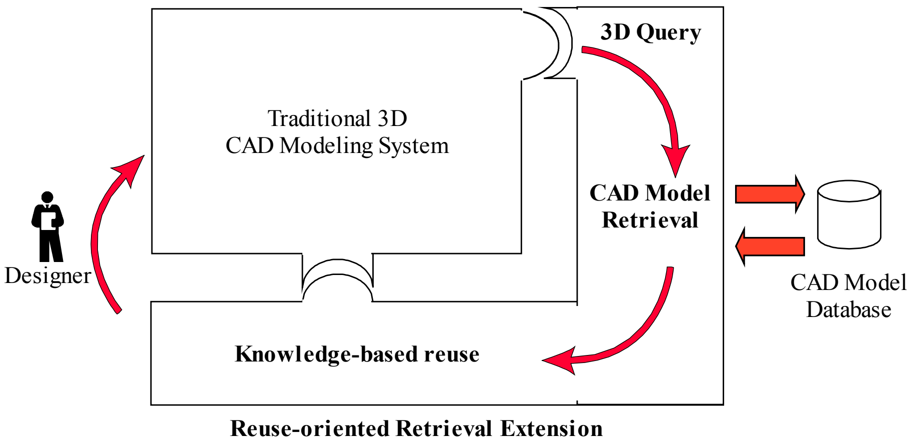


Figure 6-2. High level architecture of the prototype system

The prototype system has been realized and integrated with an industrial CAD modeling system (*i.e.*, SolidWorks) in this research. To load design files via this CAD system, CAD models are imported from native format files. Modeling knowledge and information used for ESM and PSM analysis is extracted from the CAD models by SolidWorks application programming interfaces (APIs). Geometric data can be directly obtained from the B-Rep representation of these models. Although current implementation is on a specific CAD system, the proposed approaches could be realized on any CAD systems using their own native formats, *e.g.,* .PRT files in Pro/Engineer system, because the algorithm is platform-neutral.

The workflow of ESM and PSM algorithms is illustrated in Figure 6-3. As shown in the center, the system consists of a native CAD model repository and two descriptor databases. These two descriptor databases store compact descriptor information for essential shape matching (ESM) and partial shape matching (PSM).

The top half of the figure shows the process of ESM and the bottom half illustrates the PSM process.

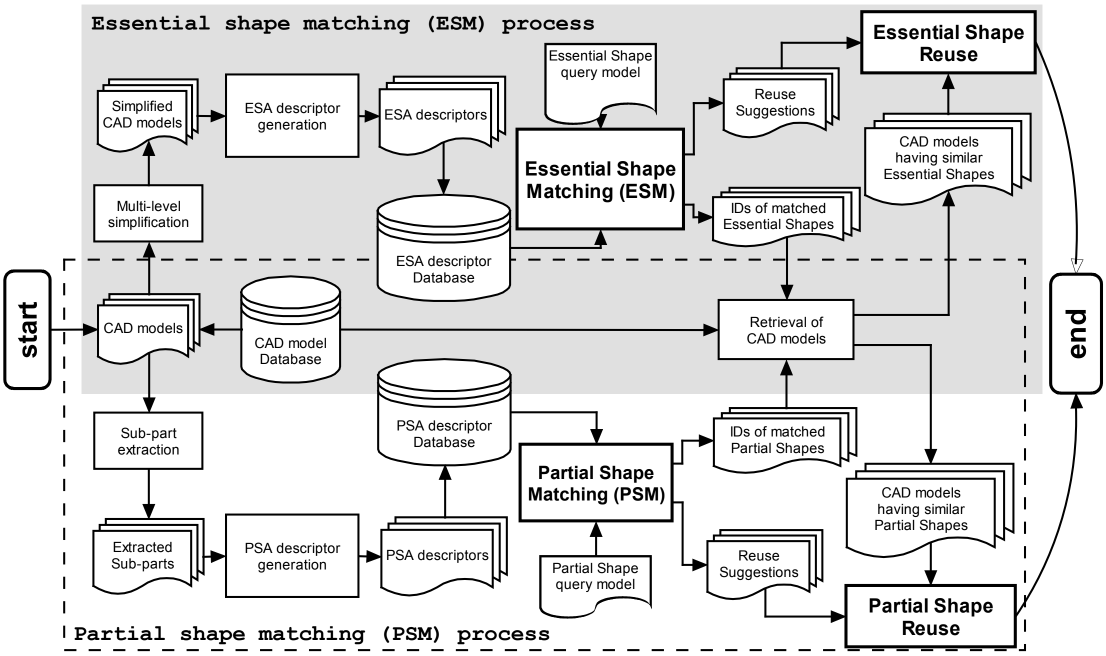


Figure 6-3. Process diagram of the prototype system

With the facilitation of the extension implemented, designers are able to finish new design tasks expeditiously by searching reusable models, without building from scratch. The design-by-reuse process usually has the following four consecutive steps, which are all well supported in the prototype implemented:

- Firstly, a designer composes a query representing what he/she wants. The prototype system allows the user to sketch basic shapes of a desirable part in a 3D way, within a popular modeling interface.
- Secondly, the designer submits the sketched query to search for similar redesign references. In this step, the prototype captures the inputted query and returns CAD models using the proposed ESM and PSM algorithms. The returned results have been assessed based on their geometric similarities and

mechanical reusabilities. The geometric similarities are compared during the ESM and PSM matching stages; and the proposed knowledge-based horizontal and vertical partitioning algorithms ensure that only the essential and partial shapes having appropriate reusabilities will be compared in matching stages.

- Thirdly, the designer freely chooses any retrieved model for reuse. During this reuse process, the prototype will also provide redesign suggestions to help designers to choose appropriate re-design modifications, which preserve the reused model fully constrained after the reuse.
- Finally, the redesigned part is automatically archived into an enterprise repository for future reuse. The system will analyze the newly reused part and generate essential and partial shape descriptors in real-time. In this way, any new part is incrementally indexed and immediately ready to future retrieval.

6.2 Evaluations on the Essential Shape Matching Algorithm

In a design repository, the parts in a design family usually share a common shape but have small detailed variants. Unlike rigid shape matching algorithms, the proposed essential shape matching (ESM) algorithm does not exactly match complete shapes. Instead, it addresses overall similarity of 3D objects while ignoring their insignificant variants. By this way, the ESM algorithm locates more reusable part varieties for the reuse-oriented retrieval. In the following sections, experiments will be conducted on realistic CAD models, to compare the retrieval effectiveness between rigid shape matching algorithms and the ESM.

6.2.1 Dataset and evaluation methods

In this research, a dataset of 629 realistic-scale CAD models has been built to test the feasibility of the proposed algorithms. These models that are designed for industrial usage, have been collected from part providers [3DContentCentral 2006]. The models are manually classified into 32 categories based on their essential shapes. The classification is shown in Figure 6-4, and all categories are listed in Table 6-1. The column proportion is the ratio of the part count of each category to the sum (*i.e.*, 629).

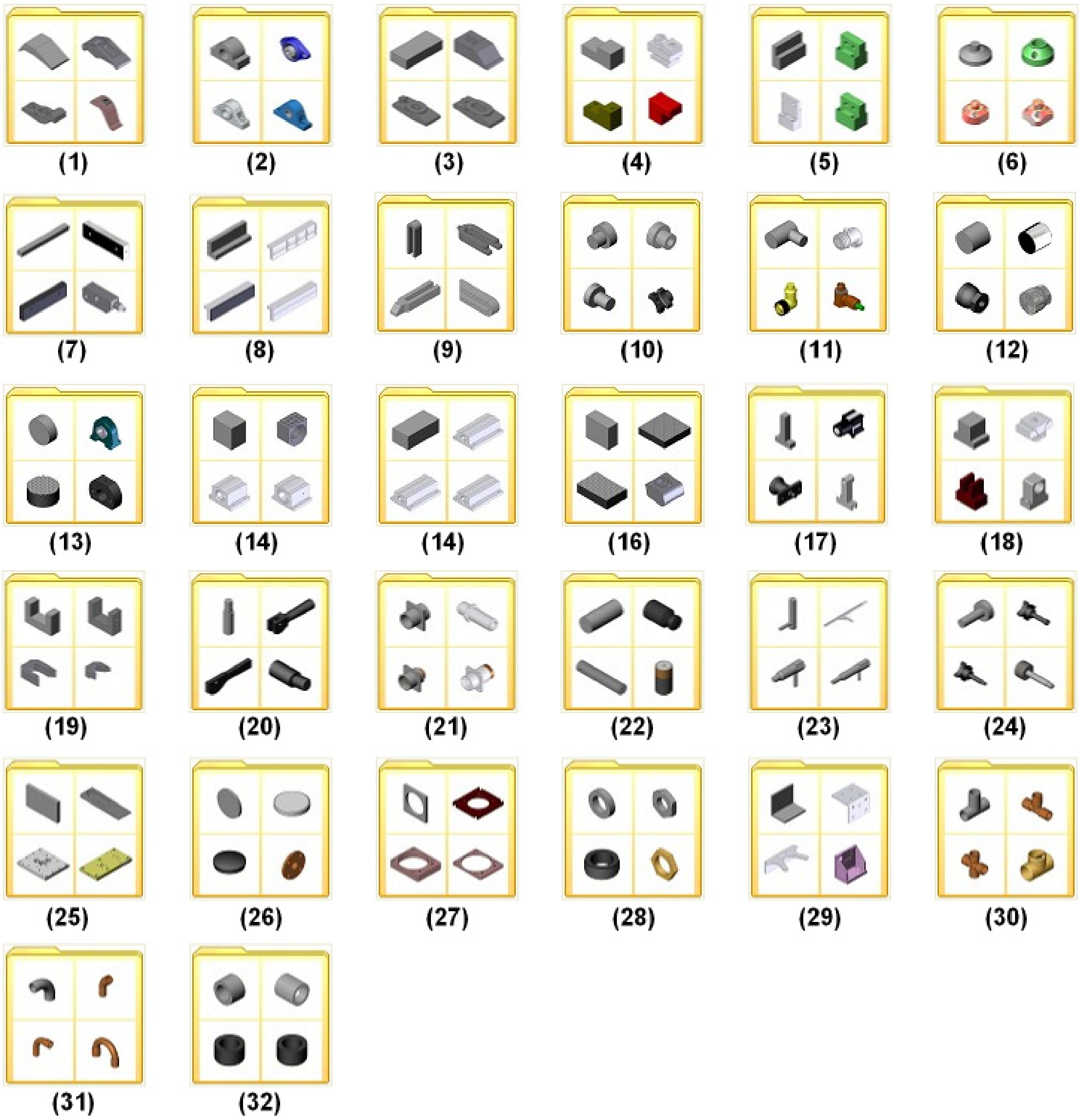


Figure 6-4. Manually classified model categories with sequence numbers

Table 6-1. Descriptions of manually classified CAD model categories

Seq	Description	Part count	Proportion
(1)	Curved flat base models	6	0.95%
(2)	Models with a rectangle- or diamond-like base and a boss	7	1.11%
(3)	Flat straight block models	45	7.15%
(4)	L-shape profile swept models with medium sweep distance	12	1.91%
(5)	L-shape profile swept models with long sweep distance	14	2.23%
(6)	Models with a flat rounded base and a boss	32	5.09%
(7)	Models with a slim and long rail shape	24	3.82%
(8)	Rail models with L-shape sweep profile	13	2.07%
(9)	U-shape swept models with two long parallel wings	4	0.64%
(10)	Swept models with a round base and a boss on the plane	33	5.25%
(11)	Models with a round base and a boss on the curved side	4	0.64%
(12)	Models with a long, rounded base	28	4.45%
(13)	Models with a short, rounded base	19	3.02%
(14)	Square block shape models	17	2.70%
(15)	Long square shape models	26	4.13%
(16)	Plain square shape models	22	3.50%
(17)	Models with a T-shape profile and a long boss	6	0.95%
(18)	Models with a T-shape profile and a short boss	27	4.29%
(19)	U-shape swept models with two short parallel wings	25	3.97%
(20)	Long pin-like shape models with small step	20	3.18%
(21)	Long pin-like shape models with locating plate	33	5.25%
(22)	Long pin-like shape models	46	7.31%
(23)	Pin-like shape models with L-shape intersection	14	2.23%
(24)	Pin-like shape models with T-shape profile	48	7.63%
(25)	Rectangle plate shape models	50	7.95%
(26)	Rounded plate shape models	6	0.95%
(27)	Rectangle plate models with holes	9	1.43%
(28)	Slim ring shape models	11	1.75%
(29)	Thin wall models with L-shape profile	7	1.11%
(30)	Tube models with an intersection	4	0.64%
(31)	Curved tube models	7	1.11%
(32)	Straight tube-like models	10	1.59%
	SUM	629	100.00%

6.2.2 Testing results and discussions

Precision (P) and recall (R) have been used regularly to measure the performance of information retrieval systems. In the experiments, precision-recall (P-R) curve is adopted for comparing the effectiveness of shape retrieval algorithms.

Recall is a measure of the ability of an algorithm to retrieve all *relevant* items, while precision is used to measure the ability to match only *relevant* items. In the manual categorization of Table 6-1, models falling into the same category are considered *relevant* in terms of essential resemblance. If {*relevant*} and {*retrieved*} represent the set of 3D objects relevant to a query and the set of 3D objects retrieved by the query respectively, the recall and precision measures are defined as:

$$Recall = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{relevant\}|} \tag{6.1}$$

$$Precision = \frac{|\{relevant\} \cap \{retrieved\}|}{|\{retrieved\}|} \tag{6.2}$$

The evaluation was made between the ESA and SD descriptors. A precision-recall curve is generated by plotting precision and recall values on a 2D graph. The precision-recall curves of different algorithms have been superimposed on the same graph to determine which algorithm is superior. The curve closer to the upper right-hand corner of the graph (where precision and recall values are maximized) indicates a better performance.

Figure 6-5a shows a sketched 3D query for searching archived locating slots. The top 5 models retrieved by ESM, shown in Figure 6-5b, are all relevant to the locating slot category; among the top 5 models retrieved by the SD D2 algorithm (see

Figure 6-5c), two are irrelevant (the second and fourth). A consequence of such irrelevancy is reflected by the P-R curves shown in Figure 6-5d: the ESM algorithm obviously outperformed the SD one as the former only evaluates the essential shape similarity while tolerating geometrically insignificant portions, consequently bringing more essential shape similar parts in its top search results. The irrelevancy brought by the SD descriptor can be explained that complex geometries of real-world models negatively affect discrimination ability of the SD descriptor, while the ESA copes with the complexity well.

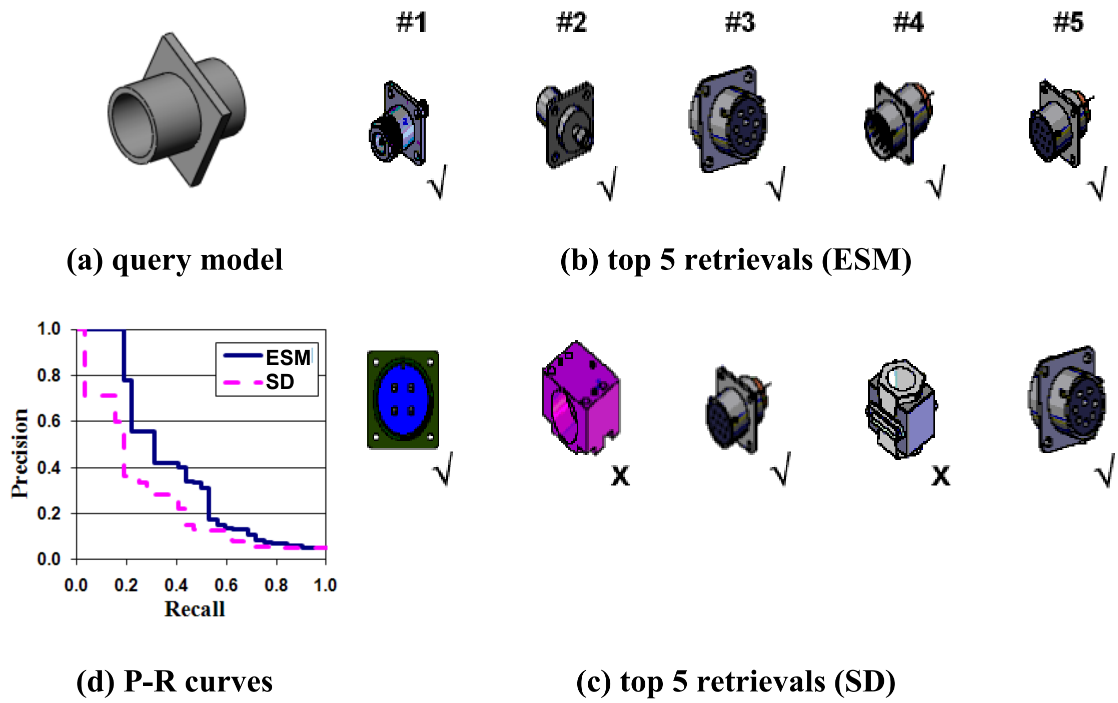


Figure 6-5. The top retrievals of ESM and SD, and the superimposed P-R curves (ESM: solid, SD: dashed)

The overall accuracy for all CAD categories is benchmarked by an average P-R curve, precision of which is an arithmetic mean of precision rates of all plotted P-R curves. The average precision mean is the sum of precisions at specified recall levels

($i.e.$, $\sum P^{\lambda}$, where P^{λ} is the precision at the recall λ, and λ is a real number of [0, 1]) divided by the count of averaged curves m (in our experiments, one P-R curve is plotted for a model category, thus m is 32) as:

$$\overline{P}_{\text{average}} = \frac{1}{m}\sum_{i=1}^{m} P_i^{\lambda} \qquad \lambda = \{0.0, 0.1, 0.2, 0.3, ..., 1.0\} \tag{6.3}$$

Figure 6-6 shows the average P-R curves of all the 32 categories, respectively. The higher curves of the ESA descriptor provides clear evidence that it performs better in retrieving broader model categories than the SD method.

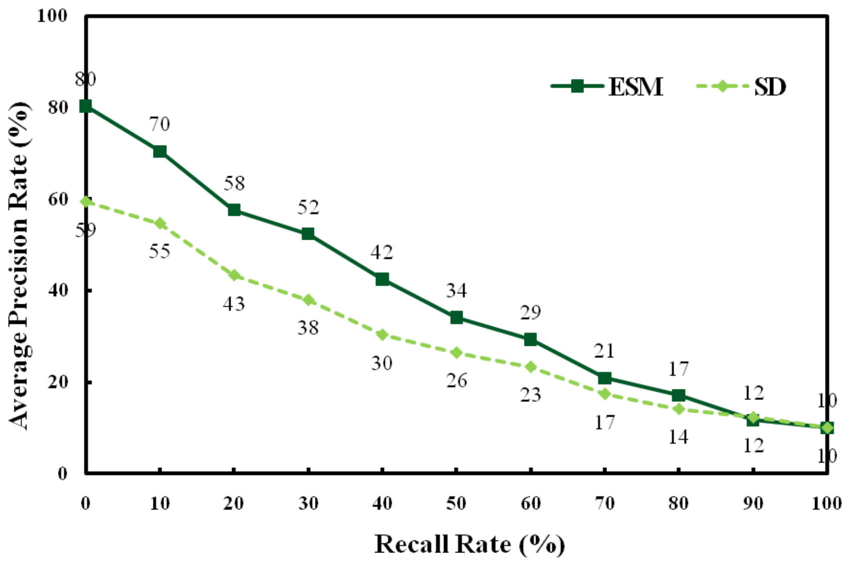


Figure 6-6. Average P-R curve comparison of ESM and SD

However, if the number of models in each category is not balanced, the simple mathematical average may be distorted because a small category of fewer models has the equivalent proportion to the average value as large categories do. To eliminate this distortion, weights of categories are introduced into the average computation as:

$$\overline{P}_{\text{weighted}} = \frac{1}{m}\sum_{i=1}^{m} w_i P_i^{\lambda} \qquad \lambda = \{0.0, 0.1, 0.2, 0.3, ..., 1.0\} \tag{6.4}$$

The weight (w_i) of a category is the model count of the category over the total number of CAD models evaluated, as listed in Table 6-1. Comparisons are further made in three recall ranges: 0 to 20%, 20 to 80%, and 80 to 100%, which correspond to high precision, middle recall, and high recall performance, respectively. Figure 6-7 reveals that the ESA descriptor is superior to the SD one in both high precision and middle recall ranges. When the recall is 10%, the ESA (73%) outperforms the SD (54%) by 19% in terms of retrieval precision. The outperformance confirms that in early search, the ESA descriptor is more efficient in relevant CAD model retrieval than rigid shape matching methods do. Discussions can also be made in the high recall range (0.8 to 1.0) as the curves of ESA and SD are more or less the same. A possible explanation is that in this range, almost all relevant models have been retrieved; therefore, the precision, which is the proportion of the relevant items of the retrieved, is expected to flatten out if the remainder is purely irrelevant models.

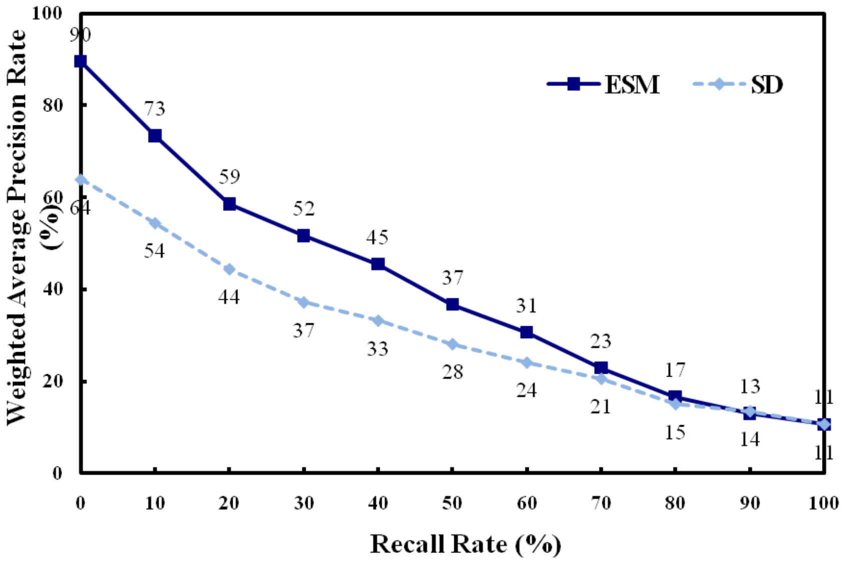


Figure 6-7. Weighted average P-R curve comparison of ESA and SD

6.2.3 Case study of essential shape matching

In this section, a case study of retrieval-oriented reuse on essential shapes is illustrated and discussed. In Figure 6-8, the process of matching a complex CAD model by a simple query is shown, which is enabled by the ESM algorithm. The figure shows a pin-connector part M and some of its simplified shapes $\{M_4, M_7, M_{10}\}$. The progressive simplification is driven by the knowledge-based horizontal FDAG partitioning algorithm. The simplified shapes are characterized by ESA descriptors, in which the geometric information is captured by a series of SD D2 histograms of the multi-level simplifications, *e.g.*, $\{H_4, H_7, H_{10}\}$ in the figure. During the essential shape matching, the D2 histogram of the query Q is compared with histograms of ESA descriptors, and the *similarity* $(M_{10}, Q) = 0.96$ is selected as the essential shape similarity of M to Q. The similarity 0.96 is also the highest similarity of all archived CAD models to the query Q.

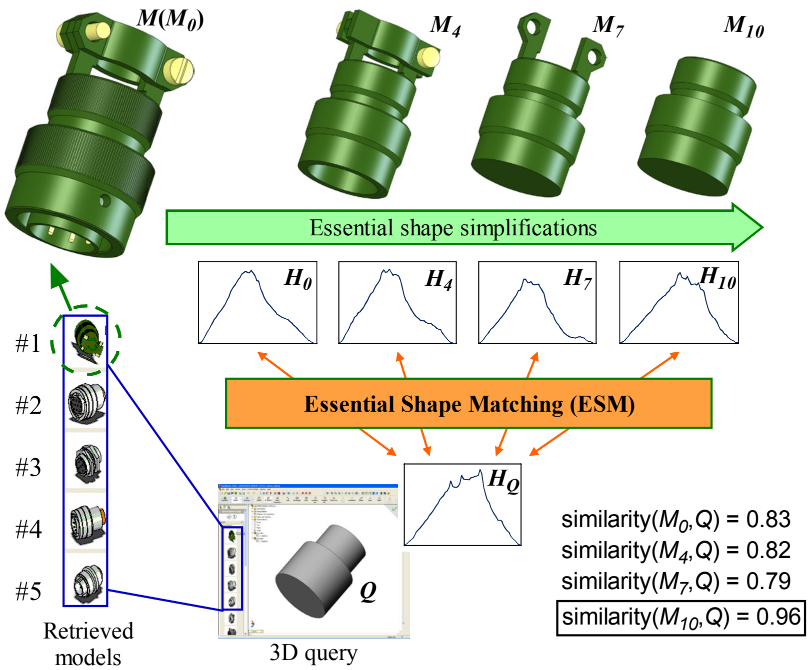


Figure 6-8. Case study of realistic CAD model retrieval enabled by ESM

Conversely, even using the same dataset and query, rigid shape matching methods cannot retrieve the M with such a simple query Q in early search. For instance, the SD D2 similarity between Q (*i.e.*, H_Q) and M (*i.e.*, H_0) is 0.83 only. A possible consequence could be that designers have to spend more time in detailing the 3D query to find out a desirable model.

Figure 6-9 illustrates more retrieval examples enabled by the proposed ESM algorithm. In the retrieval results shown on the right, all CAD models retrieved by the proposed algorithm are relevant to the sketched query. Moreover, it can be clearly observed that most of retrievals are part varieties of a same part family, which proves that retrieval and reuse by referencing another design in a part family are greatly facilitated by ESM.

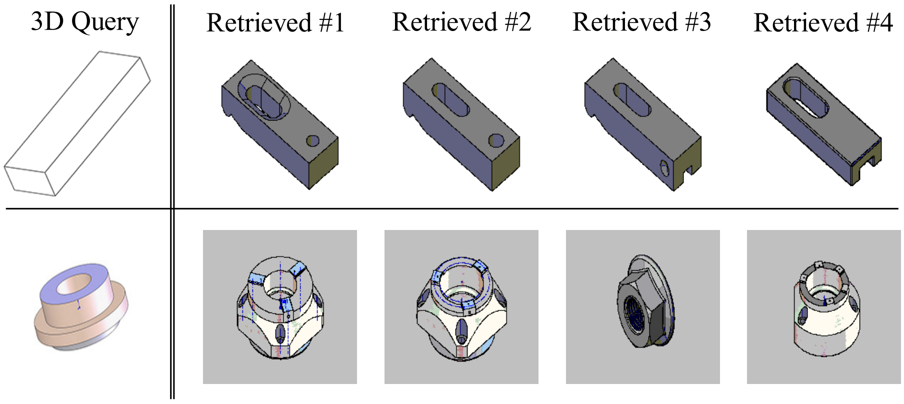


Figure 6-9. More retrieval examples enabled by the proposed essential shape matching (ESM) algorithm

6.3 Evaluations on Partial Shape Matching Algorithm

There will be a great amount of opportunities in design reuse, to create a new part by not only searching essentially similar shapes, but also migrating partial CAD components from existing parts. In Chapter 5, a knowledge-based partial shape matching (PSM) algorithm has been proposed to find out similar portions within complete CAD models. Enabled by the PSM algorithm, reusable correspondences can be recognized and located by sketched 3D query, and eventually reused in a new design. In the following sections, experiments will be conducted on real world CAD models, to compare the effectiveness of the proposed PSM algorithm in the reuse-oriented retrieval context.

6.3.1 Testing results and discussions

With the partial shape retrieval enabled by PSM algorithm, designers can easily sketch a 3D query to represent a partial shape they are looking for, and the decomposable CAD model components will be retrieved and highlighted as re-design reference. Figure 6-10 shows an example that several CAD sub-parts are located by a user-sketched query.

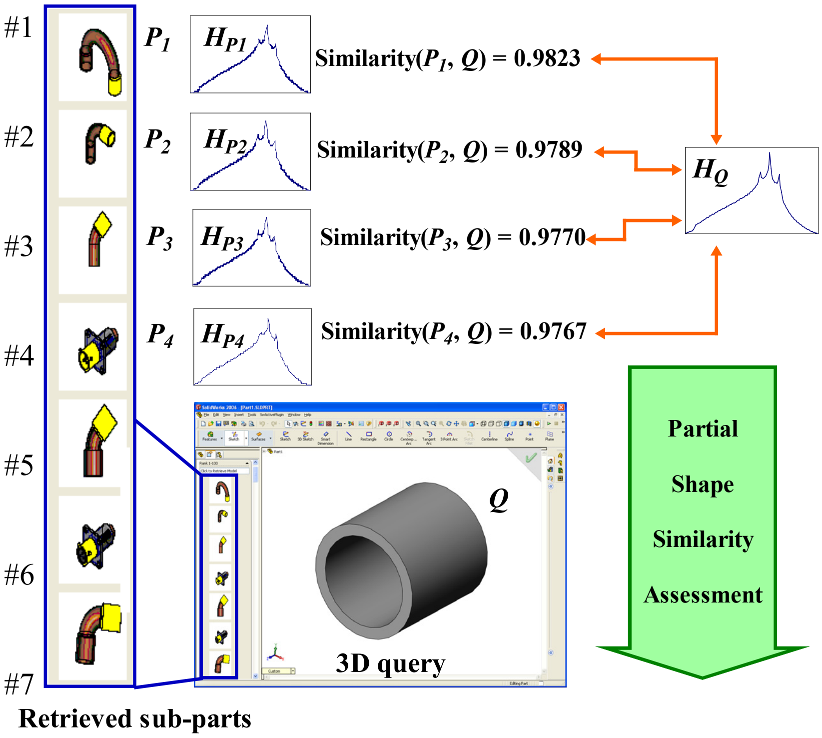


Figure 6-10. Sub-part retrieval enabled by the partial shape similarity (PSM)

In this example, the user conveniently sketched a 3D query, which specifies a desirable partial shape, shown as Q at middle-bottom of the figure. This query model is compared with sub-parts which are extracted archived complete CAD models. The partial shape similarity is assessed between Q and the segmented sub-parts to find out most similar ones. The similarity assessment is illustrated at the top-right of Figure 6-10. The highest ranked sub-parts $\{P_1, P_2, P_3, P_4\}$ (rendered in opaque) are shown on the left. The top-retrieved sub-parts closely resemble Q as their similarities to Q are all larger than 0.97; however it can be observed complete models of these sub-parts are apparently different. This example clearly demonstrates the PSM method's ability to retrieve desirable CAD components based on the partial shape similarity.

The modeling expertise embedded in archived models has also been analyzed to determine reusability of segmented sub-parts. An advantage of this knowledge-based analysis is that determining sub-parts is no longer affected by geometric factors, that is, the determination is invariable to minor changes of the shape being segmented, no matter whether surfaces have salient points, or how boundaries are surrounded by concavity. Moreover, sub-parts segmented by knowledge-based FDAG vertical partitioning are mechanically meaningful feature constitutes. Meaningful sub-parts extracted by the proposed semantic-based decomposition are shown in Figure 6-11. Dissimilar to the PSM, other methods [Bespalov *et al.* 2006, Biasotti *et al.* 2006] can barely obtain meaningful sub-parts. A possible reason is that they only extract partial shapes in a geometric way. Figure 6-12a shows sub-part correspondences determined by Reeb graph based method [Biasotti *et al.* 2006], where identical colors indicate matched partial shapes. In Figure 6-12b, partial shapes matched by a many-to-many comparison [Bespalov *et al.* 2006] are surface patches (highlighted in green colors). Such shape fragments and surface patches are meaningless in mechanical design;

therefore, they are hard to edit and even harder to reuse owning to a total lack of engineering semantics.

Figure 6-11. Mechanically meaningful sub-parts (colored in yellow) extracted by the proposed semantic-based decomposition

(a) Partial shape correspondences [Biasotti *et al.* 2006]

(b) Partially matched surfaces [Bespalov *et al.* 2006]

Figure 6-12. Less meaningful partial shapes matched by other methods

Two more examples of PSM retrieval are shown in Figure 6-13, where it can be observed that most retrieved sub-parts highlighted on the right are similar to user-specified queries shown on the left.

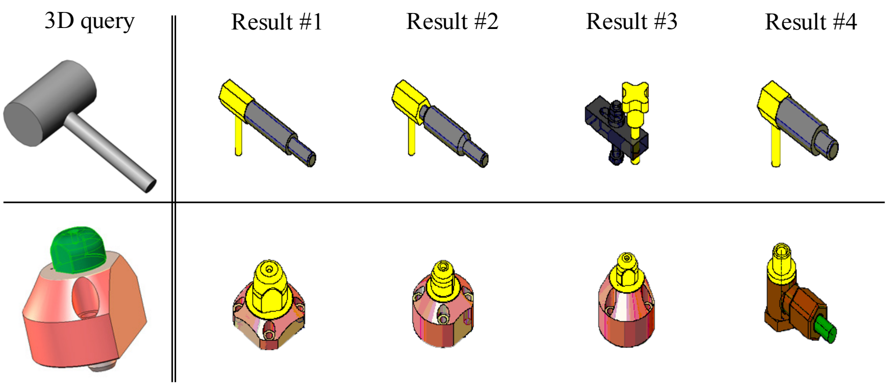


Figure 6-13. More PSM queries and retrieved results

The results appear to confirm that the proposed PSM algorithm can effectively retrieve similar sub-parts among complex CAD models. However, although the SD D2 histogram adopted in the PSA descriptor is fast in generation and comparison, it is still subject to moderate insensitivity to details. For instance, as shown in the first row of Figure 6-13, the third retrieved sub-part is not in accordance with our expectations. A possible reason of the discrepancy may be that details of this wrongly matched sub-part weaken the discriminative ability.

Furthermore, the proposed PSM algorithm provides flexible methods to compose a 3D query. The bottom row of Figure 6-13 shows a sub-part query composed of B-Rep surfaces which are specified by users (highlighted in green). This query and retrieval case clearly demonstrates that not only manifold models but also discrete surface collections can be used for finding out similar sub-parts with the proposed PSM algorithm.

6.3.2 Case study of partial shape reuse

In addition to the meaningful sub-part segmentation discussed in the last section, another advantage of the proposed PSM algorithm in reuse context is that the retrieved sub-part are feasible to externalized as a reusable components and easily portable to a new mechanical design. Once a desirable sub-part is located by the PSM algorithm, designers can reuse this sub-part easily because the knowledge analyzed in the PSM process also benefits the downstream reuse activities. In this section, a retrieval and reuse case study of a tapered head sub-part is illustrated. A design task is to create a 3D model of a locating pin based on a 2D drawing. The drawing is shown in Figure 6-15, which has a new rectangular base, but shares a standard tapered head with other existing parts. The standard sub-part will be located and reused with the facilitation of the proposed PSM algorithm, within the same CAD modeling interface.

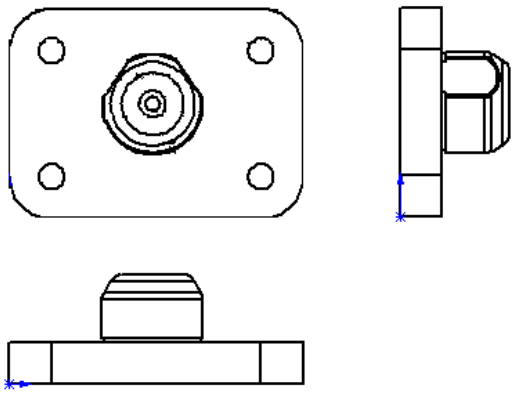


Figure 6-14. 2D drawing of a locating pin part

Because the standard tapered head sub-part can be directly copied from existing parts, designers would intuitionally create 3D sketched query similar the tapered head and perform PSM on the design repository. Figure 6-15 illustrates this sub-part reuse process. Enabled by the proposed PSM method, one can easily create a 3D query to search for desirable tapered pins (the top-middle screenshot) by a few steps of sketching. Searched by the sketched query, a number of similar sub-parts are retrieved and presented to designer, and the designer chose a retrieved sub-part as the reuse reference. The chosen model containing the sub-part is loaded within the CAD modeling system (the right middle screenshot). Figure 6-16 illustrates the major features of the loaded model in the SolidWorks modeling interface. Moreover, the matched sub-part is also automatically highlighted by the implemented prototype extension for convenient navigation and reuse.

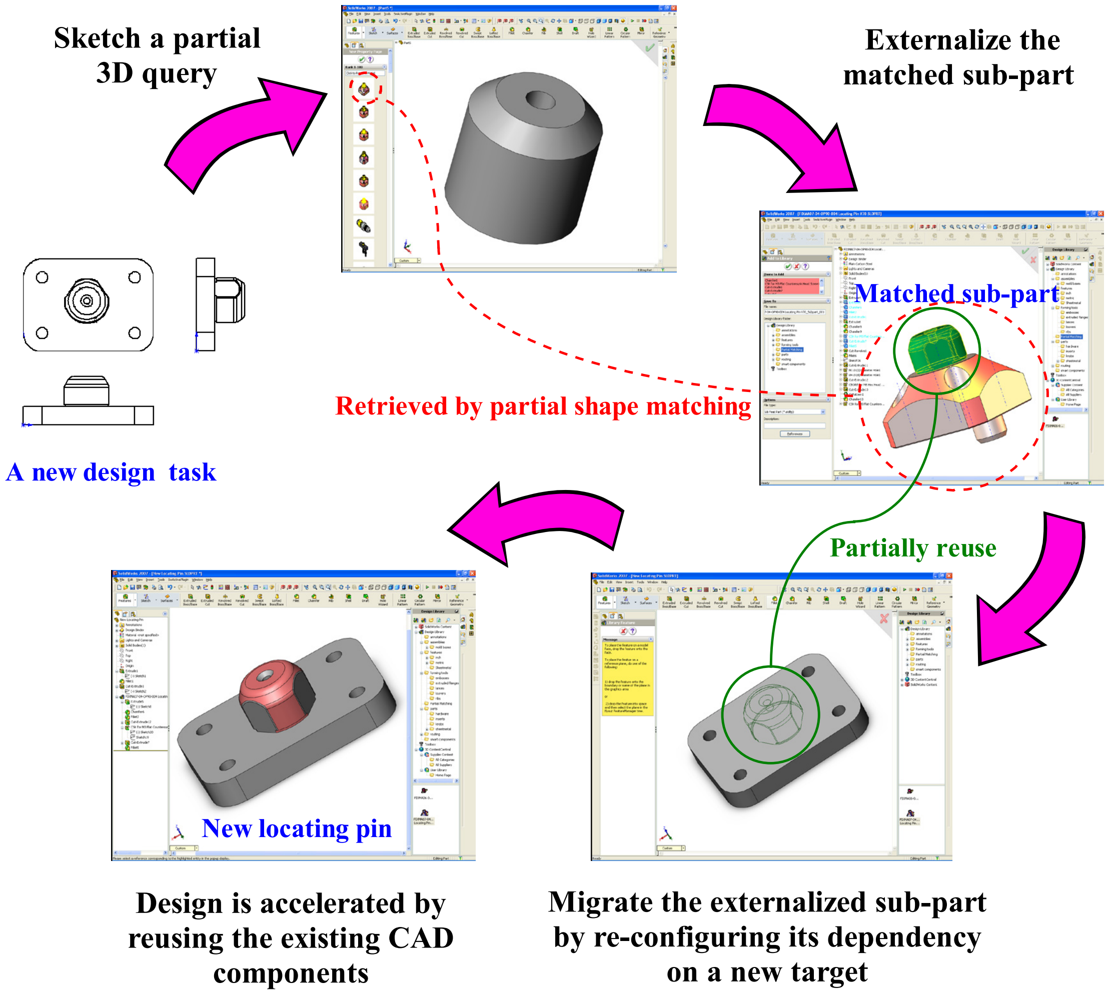


Figure 6-15. Partial shape reuse of a tapered head sub-part using PSM

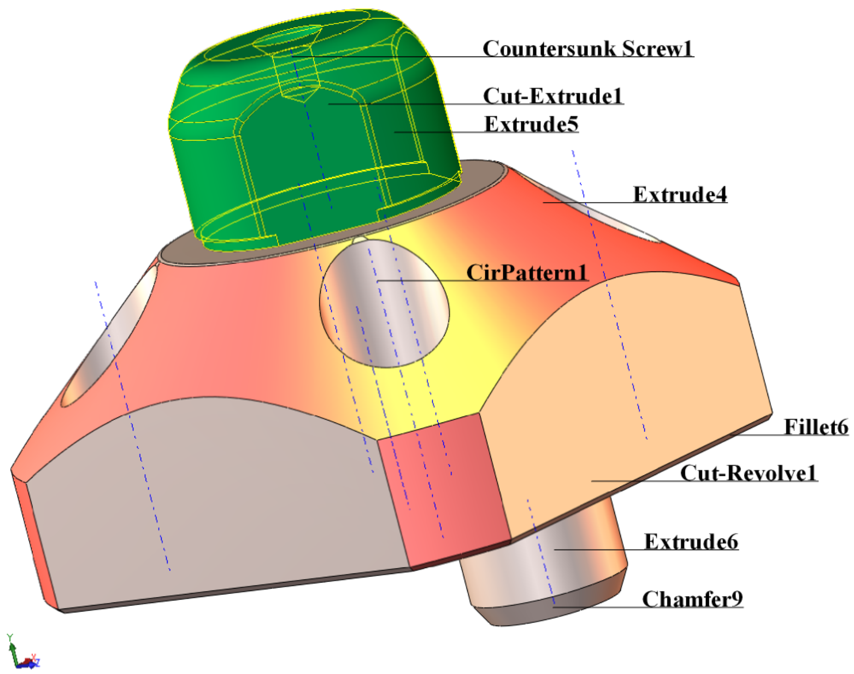


Figure 6-16. The matched sub-part and its major features.

In the meanwhile, feature modeling dependency has also been presented to the designer to facilitate decision making of the sub-part reuse. The feature modeling dependency is extracted and captured by DOT format [*Dot Language* 2009], The graph visualization in the format of Portable Network Graphics (PNG) image is automatically generated by Graphviz [2009]. The complete PNG image generated by Graphviz is shown in Figure 6-17. The highlighted feature Extrude5 is the AP of an FDAG sub-graph, which corresponds to the retrieved sub-part, *i.e.*, the standard tapered head.

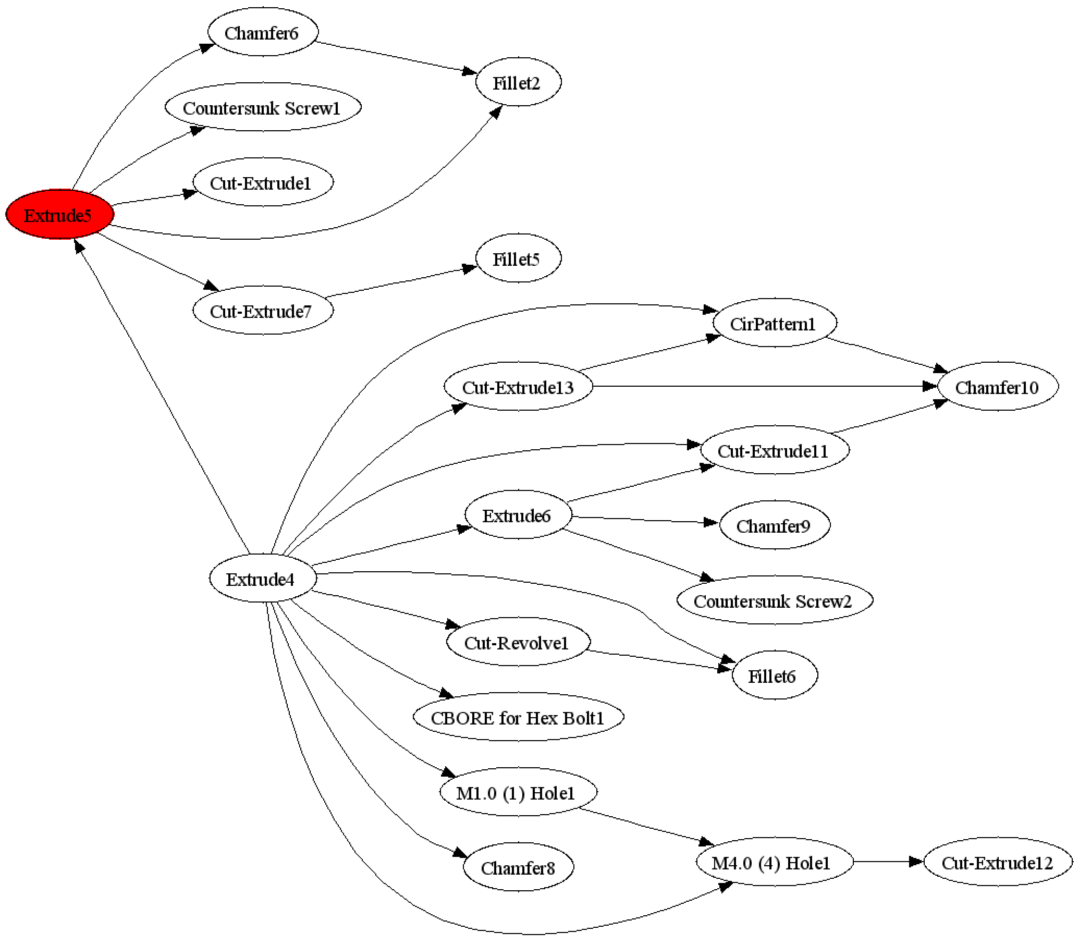


Figure 6-17. Automatically generated PNG image of the FDAG graph of the retrieved mechanical model and matched sub-part shown in Figure 6-15

The proposed sub-part partitioning algorithm analyzes external dependency of the sub-graph and deduces that the only dependency of this sub-graph is the location plane of the sub-part. Because of the limited external dependency, the matched tapered head can be conveniently externalized as a user-defined feature (UDF). Figure 6-18 highlights the externalized FDAG sub-graph.

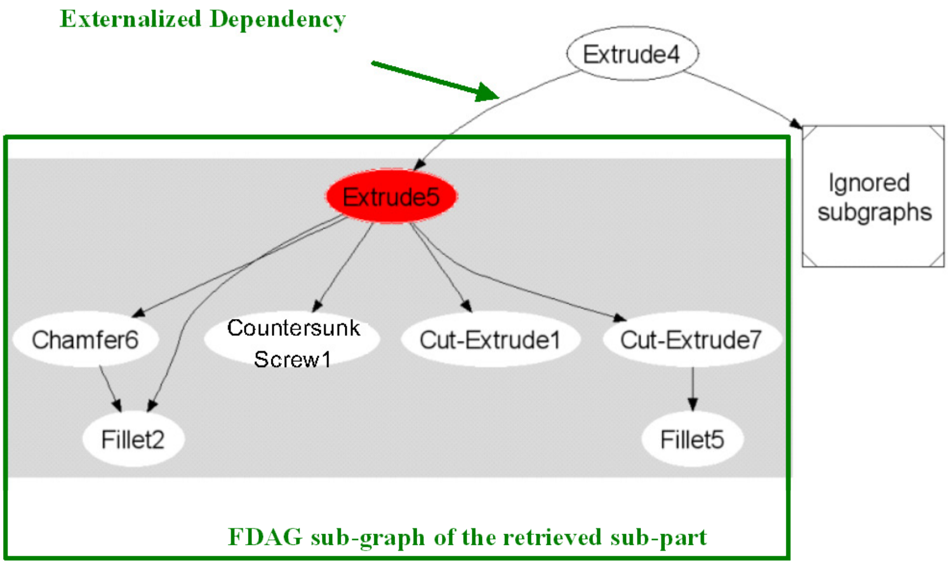


Figure 6-18. Highlighted sub-graph corresponding to the matched sub-part

On a new rectangular base, the externalized UDF can be relocated by re-configuring its locating dependency (shown at the right bottom of Figure 6-15), and eventually the retrieved sub-part is reused and becomes a part of the target design (the left bottom of the same figure). Therefore, reuse of the new locating pin is effectively accelerated by the PSM algorithm. Moreover, the reused sub-part is fully constrained and well integrated with the target; therefore, this sub-part can also be extracted, retrieved, and reused in the future.

Chapter 7 Conclusions and Recommendations

This chapter concludes the approaches presented in this thesis, discusses the contribution of research, and proposes several recommendations for future work.

7.1 Conclusions

The research investigated issues associated with the challenge of CAD model retrieval for effective design reuse. In this thesis, the modeling knowledge of CAD models has been effectively incorporated into the similarity assessment. Moreover, reuse-oriented retrieval methods have been proposed and implemented to bridge the gap between CAD model retrieval and reuse. The contributions of the research are mainly in the following areas.

(1) Elaborated a semantics-based representation for 3D CAD models to facilitate assessments on both mechanical similarity and reusability.

- An effective semantics-based FDAG representation has been proposed to capture complicated modeling interdependency among feature constitutes. Design interdependency has been examined and partially ordered modeling precedence relations have been identified for future similarity and reusability assessments. An acquisition mechanism has been put forward to extract the identified knowledge. As the extracted knowledge is a partially ordered set (POSET), a Feature Directed Acyclic Graph (FDAG) representation has been presented to convey inter-dependent structures of extracted expertise.

The FDAG representation is an organized structure that provides better views on complicated and implicit nature of design semantics.

(2) Proposed knowledge-driven FDAG partitioning schemes to extract reusable CAD components for future retrieval based on essential or partial similarity.

- Two knowledge-driven FDAG partitioning schemes have been proposed to extract reusable components. With these partitioning schemes applied on every existing model, CAD model similarity is no longer assessed as rigid 3D shapes. Instead, using the horizontal FDAG partitioning, details of models are progressively simplified; therefore, assessment on essential similarity becomes possible. On the other hand, reusable sub-parts are extracted from complete models by vertical FDAG partitioning. The partitioning schemes conform to predefined modeling constraints. Hereby, extracted components are highly reusable in terms of variant and adaptive design, and inflexibility to reuse can be greatly avoided.

(3) Proposed an approach to support effective retrieval of CAD models based on their essential shape similarities.

- An approach supporting CAD model retrieval based on their essential shape similarities has been presented. In order to compare model similarity on essential shapes, a horizontal partitioning scheme has been put forward to decompose FDAG graph from minimal elements to maximal ones. With the FDAG decomposition, a full-detailed CAD model will be simplified progressively. Each simplification has fewer details compared with the previous one while maintaining the essential shape of the model. Furthermore, generation algorithms of essential shape descriptors have been formalized in CAD modeling context, and essential shape similarity has also been defined. Based on the defined similarity, the essential shape matching (ESM) has been given to perform essential similarity assessment and retrieval.

(4) Put forward a method to support effective retrieval of reusable CAD model components based on their partial shape similarities.

- A method to support effective retrieval of reusable CAD model components based on their partial shape similarities has been put forward. In order to serve partial similarity assessment, a vertical partitioning scheme has been applied to find out disjointed sub-graph from FDAG representation, by examining the reachability of a POSET data. The found disjointed sub-graphs are equivalent to sub-parts of a mechanical design. Furthermore, partial shape descriptors have been defined, and partial similarity was compared with sub-part level. Based on the partial similarity, a partial shape matching (PSM) algorithm has been presented to address partial similarity assessment and retrieval.

(5) Developed a prototype system to support the proposed reuse-oriented retrieval, which can locate reusable CAD models on their essential or partial similarity.

- The reuse-oriented retrieval system requirements have been examined to ensure the gap between retrieval and reuse is fulfilled. A prototype system has been successfully implemented to demonstrate the feasibility of the proposed methods. The effectiveness has also been evaluated using over six hundred realistic CAD models. The results showed that the proposed method outperforms other methods. In addition, a query interface is embedded in a modeling system, which allows users to seamlessly compose query when designing. Moreover, query results can be proactively promoted to designers based on the 3D query sketched by designers, to significantly shorten the process from querying to retrieving. Lastly, in retrieval stage, redesign suggestions can be visualized to help users choose the most feasible re-design options. This keeps the reused design still parametrically constrained, which is also assessable for future similarity comparison.

The research also clearly demonstrates several desired advantages of the proposed reuse-oriented retrieval. Firstly, the reuse-oriented retrieval paradigm allows designers to retrieve reusable models by specifying an essential or partial shape query. In this way, designers can retrieve essentially similar parts (*e.g.*, a part family) or meaningful sub-parts as redesign references at any time of design process. Secondly, it offers ease of use during reuse of retrieved results, because modeling dependency analysis is done in the retrieval stage to ensure the reusability of the retrieved. Therefore, design reuse inflexibility can be effectively avoided. Thirdly, it helps preservation of design intelligence to newly reused design. The implemented prototype system provides designers access to original design expertise embedded in CAD models by visualizing FDAG graphs during the reuse stage. As a result, the best design practices in original parts, including but not limited to modeling precedence and parametric constraints, are inherently transferred to new designs.

7.2 Recommendations for Future Work

While the reuse-oriented retrieval method developed in this thesis can assist engineering designers in retrieval reusable CAD models in the development of new products, there are still several research opportunities for further improvement and investigation.

7.2.1 Extension to support cross-system retrieval

Cross-system retrieval possibly is required as designers sometimes need to retrieve CAD models from a system consisting of heterogeneous CAD file formats. In this research, the FDAG extraction is system-specific as it involves reading modeling knowledge and implications from native file formats. However, the subsequent algorithms, *i.e.*, the FDAG decomposition, and shape descriptor generation, are both neutral to CAD systems. In order to deal with models sourcing from heterogeneous CAD systems, more studies need to be conducted on a neutral feature modeling knowledge representation. For instances, representing feature dependency in a neutral way could be a possible research direction. In a study on heterogeneous CAD data exchange [Chen *et al.* 2006], procedural models of different CAD systems are exchanged via a series of neutral modeling commands [Li *et al.* 2007]. The NMC-based models can be used to extract a neutral representation of feature dependency, which then can be converted to an FDAG representation for cross-system retrieval.

This can integrate more design reference sources into prototype system and revitalize more existing mechanical parts for future design.

7.2.2 Extension to support cross-system reuse

Another possibility is that a retrieved model in the format of one CAD modeling system could be reused in another CAD system. This is the cross-system reuse. Although our implementation can be devised for cross-system retrieval, it is difficult to reuse one retrieved component on another heterogeneous model as constrained B-rep entities may be missing in the target model. Differences in modeling kernels could cause that the same feature has identical shapes but differs in B-rep topologies and definitions among heterogeneous CAD systems [Rappoport *et al.* 2005]. This is a limitation of history based modeling. In order to address this gap, direct modeling, which allows direct geometry modification instead of feature modeling, could be adopted in the stage of redesign, especially for reuse of complex aerospace parts [Jackson and Buxton 2007]. Therefore, more studies should be conducted in this area about how to effectively support the direct modeling on retrieved models.

7.2.3 Integration of part classification view

A classification view of all existing part could be integrated into the reuse-oriented retrial framework to accelerate selection of redesign references. In this thesis, one-dimensional view of all CAD models can be obtained based on the general or partial shape similarity to the sketched query model. However, without inputting a sketched query, designers may need to have an overall navigation view of all parts and choose the desirable one. In the navigation that could be a two or three-dimensional view, the distance between pairs of displayed part corresponds to the proposed shape similarity metrics. Moreover, for a database with thousands of 3D models, clustering technique could be implemented to provide hierarchical views.

Publications

- Li, M., Fuh, J. Y. H., Zhang, Y. F. and Qiu, Z. M., 2009, "Towards effective mechanical design reuse: feature-based CAD model retrieval on general shapes and partial shapes", ASME Transactions - Journal of Mechanical Design, Vol. 131, No. 12, pp. 124501.
- Li, M., Zhang, Y. F. and Fuh, J. Y. H., 2010, "Retrieving reusable 3D CAD Models Using Knowledge-Driven Dependency Graph Partitioning", Computer-Aided Design and Applications, Vol. 7, No. 3, pp. 417-430.
- Li, M., Zhang, Y. F., Fuh, J. Y. H. and Qiu, Z. M., 2011, "Design reusability assessment for effective CAD model retrieval and reuse", International Journal of Computer Applications in Technology (IJCAT), Inder Science Publisher, Vol. 40, No.1/2, pp. 3-12.
- Li, M., Gao, S. M., Fuh, J. Y. H. and Zhang, Y. F., 2008, "Replicated Concurrency Control for Collaborative Feature Modelling: A Fine Granular Approach", Computers in Industry, Vol. 59, No. 9, pp. 873-881.
- Li, M., Fuh, J. Y. H., Zhang, Y. F. and Qiu, Z. M., 2008, "General and partial shape matching approaches on feature-based CAD models to support efficient part retrieval", Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, New York, United States, August 3-6.
- Li, M., Fuh, J. Y. H., Zhang, Y. F. and Gao, S., 2008, "Adaptive Granular Concurrency Control for Replicated Collaborative Feature Modeling", Proceedings of 12th International Conference on Computer Supported Cooperative Work in Design, Xian, China, April 16-18, Vol. 1, pp. 116-122.
- Li, M., Fuh, J. Y. H., Zhang, Y. F. and Qiu, Z. M., 2007, "A Multi-Level Reusability Assessment Mechanism for Parametric Mechanical Design", Proceedings of 2007 International Conference on Product Design and Manufacturing Systems, PDMS 2007, Chongqing, China, October 12-15.
- Li, M., Gao, S. M., Fuh, J. Y. H. and Zhang, Y. F., 2007, "A Fine Granular Concurrency Control Mechanism for a Peer-to-Peer Cooperative Design

Environment", Proceedings of 11th International Conference on Computer Supported Cooperative Work in Design, Melbourne, Australia, April 26-28, pp. 180-185.

References

3DContentCentral 2006, *Free Supplier Parts Models, CAD Drawings, Assembly Models*, <http://www.3dcontentcentral.com/>.

Agathos, A., Pratikakis, I., Perantonis, S., Sapidis, N. and Azariadis, P. 2007, 3D Mesh Segmentation Methodologies for CAD Applications, *Computer-Aided Design and Applications*, vol. 4, no. 1-6, pp. 827-841.

Akgul, C. B., Sankur, B., Yemez, Y. and Schmitt, F. 2007, Density-Based 3D Shape Descriptors, *Eurasip Journal on Advances in Signal Processing*, p. 32503.

Ankerst, M., Kastenmuller, G., Kriegel, H. P. and Seidl, T. 1999, 3D Shape Histograms for Similarity Search and Classification in Spatial Databases, *Proceedings of the 6th International Symposium on Advances in Spatial Databases*, Hong Kong, China, July 20-23, pp. 207–226.

Ansary, T. F., Daoudi, M. and Vandeborre, J.-P. 2007, A Bayesian 3-D Search Engine Using Adaptive Views Clustering, *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 78-88.

Bai, J., Liu, Y. and Gao, S. 2007, Multi-Mode Solid Model Retrieval Based on Partial Matching, *Proceedings of 10th IEEE International Conference on Computer Aided Design and Computer Graphics, CAD/Graphics 2007*, Beijing, China, October 15-18, pp. 126-131.

Barton, J. and Love, D. 2005, Retrieving Designs from a Sketch Using an Automated Gt Coding and Classification System, *Production Planning and Control*, vol. 16, no. 8, pp. 763–773.

Bespalov, D., Regli, W. C. and Shokoufandeh, A. 2003a, Reeb Graph Based Shape Retrieval for CAD, *Proceedings of 2003 ASME Design Engineering Technical Conferences, DETC'03*, Chicago, United States, September 2-6, vol. 1, pp. 229–238.

Bespalov, D., Shokoufandeh, A., Regli, W. C. and Sun, W. 2003b, Scale-Space Representation and Classification of 3D Models, *Journal of Computing and Information Science in Engineering*, vol. 3, no. 4, pp. 315–324.

Bespalov, D., Ip, C. Y., Regli, W. C. and Shaffer, J. 2005, Benchmarking CAD Search Techniques, *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, Cambridge, Massachusetts, United States, June 13-15, pp. 275–286.

Bespalov, D., Regli, W. C. and Shokoufandeh, A. 2006, Local Feature Extraction and Matching Partial Objects, *Computer-Aided Design*, vol. 38, no. 9, pp. 1020–1037.

Bettig, B. and Shah, J. 2001, Derivation of a Standard Set of Geometric Constraints for Parametric Modeling and Data Exchange, *Computer Aided Design*, vol. 33, no. 1, pp. 17-33.

Biasotti, S., Marini, S., Spagnuolo, M. and Falcidieno, B. 2006, Sub-Part Correspondence by Structural Descriptors of 3D Shapes, *Computer-Aided Design*, vol. 38, no. 9, pp. 1002–1019.

Cardone, A., Gupta, S. K. and Karnik, M. 2003, A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications, *Journal of Computing and Information Science in Engineering*, vol. 3, no. 2, pp. 109-118.

Cardone, A., Gupta, S. K. and Karnik, M. 2004, Identifying Similar Parts for Assisting Cost Estimation of Prismatic Machined Parts, *Proceedings of the ASME Design Engineering Technical Conference 2004*, Salt Lake City, United States, vol. 3, pp. 765–777.

Cardone, A., Gupta, S. K., Deshmukh, A. and Karnik, M. 2006, Machining Feature-Based Similarity Assessment Algorithms for Prismatic Machined Parts, *Computer-Aided Design*, vol. 38, no. 9, pp. 954–972.

Chang, T.-C. and Chen, S.-Y. 1999, Character Segmentation Using Convex-Hull Techniques, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, no. 6, pp. 833-858.

Chen, D., Tian, X. P., Shen, Y. T. and Ouhyoung, M. 2003, On Visual Similarity Based 3D Model Retrieval, *Proceedings of 24th Annual Conference EUROGRAPHICS - Computer Graphics Forum*, Granada, Spain, September 1-5, vol. 22, pp. 223–232.

Chen, D. Y. and Ouhyoung, M. 2002, A 3D Object Retrieval System Based on Multi-Resolution Reeb Graph, *Proceedings of Computer Graphics Workshop 2002*, Taiwan, June 2002, p. 16.

Chen, X., Li, M. and Gao, S. 2006, A Web Service for Exchanging Procedural CAD Models between Heterogeneous CAD Systems, in W. Shen, K.-M. Chao, Z. Lin, J.-P. A. Barthès and A. E. James (eds), *Computer Supported Cooperative Work in Design Ii*, Springer Verlag, Heidelberg, Germany, vol. 3865 LNCS, pp. 225–234.

Cheng, H.-C., Chu, C.-H., Wang, E. and Kim, Y.-S. 2007, D Part Similarity Comparison Based on Levels of Detail in Negative Feature Decomposition Using Artificial Neural Network, *Computer-Aided Design and Applications*, vol. 4, no. 5, pp. 619-628.

Chu, C.-H. and Hsu, Y. C. 2006, Similarity Assessment of 3D Mechanical Components for Design Reuse, *Robotics and Computer-Integrated Manufacturing*, vol. 22, no. 4, pp. 332–341.

Cicirello, V. A. and Regli, W. C. 2002, An Approach to a Feature-Based Comparison of Solid Models of Machined Parts, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 16, no. 5, pp. 385-399.

Cormen, T., Leiserson, C., Rivest, R. and Stein, C. 2001, *Introduction to Algorithms*, The MIT Press.

Cyr, C. M. and Kimia, B. B. 2001, 3D Object Recognition Using Shape Similiarity-Based Aspect Graph, *Proceedings of 8th IEEE International Conference on Computer Vision (ICCV 2001)*, Vancouver, Canada, July 7-14 vol. 1, pp. 254-261.

Deo, N. 1974, *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall Series in Automatic Computation, Prentice Hall.

, *Dot Language* 2009, <http://en.wikipedia.org/wiki/DOT_language>.

El-Mehalawi, M. and Miller, R. A. 2003a, A Database System of Mechanical Components Based on Geometric and Topological Similarity. Part I: Representation, *Computer-Aided Design*, vol. 35, no. 1, pp. 83–94.

El-Mehalawi, M. and Miller, R. A. 2003b, A Database System of Mechanical Components Based on Geometric and Topological Similarity. Part Ii: Indexing, Retrieval, Matching, and Similarity Assessment, *Computer-Aided Design*, vol. 35, no. 1, pp. 95–105.

Elad, M., Tal, A. and Ar, S. 2001, Content Based Retrieval of Vrml Objects: An Iterative and Interactive Approach, *Proceedings of the sixth Eurographics workshop on Multimedia 2001*, Manchester, UK, pp. 107-118.

Floyd, R. W. 1962, Algorithm 97: Shortest Path, *Communications of the ACM*, vol. 5, no. 6, p. 345.

Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D. and Jacobs, D. 2003, A Search Engine for 3D Models, *ACM Transactions on Graphics*, vol. 22, no. 1, pp. 83–105.

Gal, R., Shamir, A. and Cohen-Or, D. 2007, Pose-Oblivious Shape Signature, *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 261-271.

Gao, W., Gao, S. M., Liu, Y. S., Bai, J. and Hu, B. K. 2006, Multiresolutional Similarity Assessment and Retrieval of Solid Models Based on Dbms, *Computer-Aided Design*, vol. 38, no. 9, pp. 985–1001.

García, F. G., Feixas, M. and Sbert, M. 2007, View-Based Shape Similarity Using Mutual Information Spheres, *The Annual Conference of the European Association for Computer Graphics, EUROGRAPHICS 2007*, Prague, Czech Republic, September 3-7.

Graphviz 2009, *Graphviz*, <http://graphviz.org/>.

Gunn, T. G. 1982, The Mechanization of Design and Manufacturing, *Scientific American*, vol. 247, no. 3, pp. 86–88.

Gupta, S. K., Cardone, A. and Deshmukh, A. 2006, Content-Based Search Techniques for Searching CAD Databases, *Computer-Aided Design and Applications*, vol. 3, no. 6, pp. 811-819.

Herrmann, J. W. and Singh, G. 1997, *Design Similarity Measures for Process Planning and Design Evaluation*, TR97-74, Institute of Systems Research, University of Maryland, College Park, College Park MD.

Hilaga, M., Shinagawa, Y., Kohmura, T. and Kunii, T. L. 2001, Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes, *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, United States, pp. 203-212.

Hong, T., Lee, K., Kim, S., Chu, C. and Lee, H. 2005, Similarity Comparison of Mechanical Parts, *Computer-Aided Design and Applications*, vol. 2, no. 6, pp. 759–768.

Hong, T., Lee, K. and Kim, S. 2006, Similarity Comparison of Mechanical Parts to Reuse Existing Designs, *Computer-Aided Design*, vol. 38, no. 9, pp. 973–984.

Hopcroft, J. and Tarjan, R. 1973, Algorithm 447: Efficient Algorithms for Graph Manipulation, *Communications of the ACM*, vol. 16, no. 6, pp. 372-378.

Hou, S. and Ramani, K. 2008, Structure-Oriented Contour Representation and Matching for Engineering Shapes, *Computer-Aided Design*, vol. 40, no. 1, pp. 94-108.

Hou, X., Zhang, X. and Liu, W. 2007, Using Enhanced Shape Distributions to Compare CAD Models, in *Advances in Multimedia Information Processing – Pcm 2007*, Springer Berlin / Heidelberg, vol. 4810/2007, pp. 385-388.

Ip, C. Y., Lapadat, D., Sieger, L. and Regli, W. C. 2002, Using Shape Distributions to Compare Solid Models, *Proceedings of the seventh ACM symposium on Solid modeling and applications*, Saarbrucken, Germany, June 17-21, pp. 273–280.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y. and Ramani, K. 2005a, Shape-Based Searching for Product Lifecycle Applications, *Computer-Aided Design*, vol. 37, no. 13, pp. 1435-1446.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y. and Ramani, K. 2005b, Three-Dimensional Shape Searching: State-of-the-Art Review and Future Trends, *Computer-Aided Design*, vol. 37, no. 5, pp. 509–530.

Iyer, S. and Nagi, R. 1994, Identification of Similar Parts in Agile Manufacturing, *American Society of Mechanical Engineers, Design Engineering Division (Publication) DE*, Chicago, United States, vol. 74, pp. 87–96.

Jackson, C. and Buxton, M. 2007, *The Design Reuse Benchmark Report - Seizing the Opportunity to Shorten Product Development*, 3908, Aberdeen Group, Boston, United States.

Jaume, S., Macq, B. and Warfield, S. K. 2002, Labeling the Brain Surface Using a Deformable Multiresolution Mesh, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002*, vol. 5, pp. 451-458.

Ju, T., Baker, M. L. and Chiu, W. 2007, Computing a Family of Skeletons of Volumetric Models for Shape Description, *Computer-Aided Design*, vol. 39, no. 5, pp. 352–360.

Katz, S. and Tal, A. 2003, Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts, *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 954-961.

Kazhdan, M., Chazelle, B., Dobkin, D., Funkhouser, T. and Rusinkiewicz, S. 2004, A Reflective Symmetry Descriptor for 3D Models, *Algorithmica*, vol. 38, no. 1, pp. 201-225.

Kim, S., Lee, K., Hong, T., Kim, M., Jung, M. and Song, Y. 2005, An Integrated Approach to Realize Multi-Resolution of B-Rep Model, *ACM Symposium on Solid Modeling and Applications, SM*, Cambridge, MA, United States, pp. 153–162.

Klasing, K., Wollherr, D. and Buss, M. 2008, A Clustering Method for Efficient Segmentation of 3D Laser Data, *International Conference on Robotics and Automation - ICRA 2008*, Pasadena, California, USA, May 19-23, 2008, pp. 4043-4048.

Klette, G. and Pan, M. 2005, 3D Topological Thinning by Identifying Non-Simple Voxels, in *Combinatorial Image Analysis*, Springer, vol. LNCS 3322, pp. 164-175.

Kriegel, H. P., Kroger, P., Mashael, Z., Pfeifle, M., Potke, M. and Seidl, T. 2003, Effective Similarity Search on Voxelized CAD Objects, *Proceedings 8th International Conference on Database Systems for Advanced Applications*, Kyoto, Japan, pp. 27–36.

Lakare, S. 2000, *3D Segmentation Techniques for Medical Volumes*, Center for Visual Computing, Department of Computer Science, Stony Brook University, New York, USA.

Lee, S. Y. and Fischer, G. W. 1999, Grouping Parts Based on Geometrical Shapes and Manufacturing Attributes Using a Neural Network, *Journal of Intelligent Manufacturing*, vol. 10, no. 2, pp. 199–209.

Leifman, G., Meir, R. and Tal, A. 2005, Semantic-Oriented 3D Shape Retrieval Using Relevance Feedback, *Visual Computer*, vol. 21, no. 8-10, pp. 865-875.

Li, M., Gao, S. M. and Wang, C. C. L. 2007, Real-Time Collaborative Design with Heterogeneous CAD Systems Based on Neutral Modeling Commands, *Journal of Computing and Information Science in Engineering*, vol. 7, no. 2, pp. 113–125.

Li, X. 2010, A New Clustering Segmentation Algorithm of 3D Medical Data Field Based on Data Mining, *International Journal of Digital Content Technology and its Applications*, vol. 4, no. 4, pp. 174-181.

Liu, R. and Zhang, H. 2004, Segmentation of 3D Meshes through Spectral Clustering, paper presented to Proceedings of 12th Pacific Conference on Computer Graphics and Applications, Seoul, South Korea, October 6-8, <http://www.pdffinder.com/get/segmentation-of-3d-meshes-through-spectral-clustering.pdf>.

Lou, K., Ramani, K. and Prabhakar, S. 2004, Content-Based Three-Dimensional Engineering Shape Search, *Proceedings of 20th International Conference on Data Engineering, ICDE04*, Boston, United States, March 30 - April 2, pp. 754–765.

Lou, K., Iyer, N., Jayanti, S., Kalyanaraman, Y., Prabhakar, S. and Ramani, K. 2005, Effectiveness and Efficiency of Three-Dimensional Shape Retrieval, *Journal of Engineering Design*, vol. 16, no. 2, pp. 175-194.

Love, D. and Barton, J. 2001, Drawing Retrieval Using an Automated Coding Technique, *Proceedings of 11th International Conference on Flexible Automation 2001*, Dublin, July, pp. 158-166.

Love, D. and Barton, J. 2004, Aspects of Design Retrieval Performance Using Automatic Gt Coding of 2d Engineering Drawings, *5th International Conference on Integrated Design and Manufacturing in Mechanical Engineering (IDMME 2004)*, Bath, UK, p. 46.

Mantyla, M., Nau, D. and Shah, J. J. 1996, Challenges in Feature-Based Manufacturing Research, *Communications of the ACM*, vol. 39, no. 2, pp. 77-85.

Mäntylä, M., Nau, D. and Shah, J. 1996, Challenges in Feature-Based Manufacturing Research, *Communications of the ACM*, vol. 39, no. 2, pp. 77-85.

McWherter, D., Peabody, M., Regli, W. C. and Shokoufandeh, A. 2001, Solid Model Databases: Techniques and Empirical Results, *Journal of Computing and Information Science in Engineering*, vol. 1, no. 4, pp. 300-310.

Miller, E. 1998, What Is Pdm?, *Mechanical Engineering*, October.

Min, P., Kazhdan, M. and Funkhouser, T. 2004, A Comparison of Text and Shape Matching for Retrieval of Online 3D Models, *Proceedings of 8th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2004.* , Bath, United Kingdom, vol. LNCS 3232, pp. 209–220.

Mitrofanov, S. P. 1966, *The Scientific Principles of Group Technology*, National Lending Library for Science and Technology, Boston Spa, Yorkshire, United Kingdom.

Morris, R. J., Najmanovich, R. J., Kahraman, A. and Thornton, J. M. 2005, Real Spherical Harmonic Expansion Coefficients as 3D Shape Descriptors for Protein Binding Pocket and Ligand Comparisons, *Bioinformatics*, vol. 21, no. 10, pp. 2347–2355.

Ohbuchi, R., Minamitani, T. and Takei, T. 2005, Shape-Similarity Search of 3D Models by Using Enhanced Shape Functions, *International Journal of Computer Applications in Technology*, vol. 23, no. 2-4, pp. 70-85.

Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D. 2001, Matching 3D Models with Shape Distributions, *Proceedings of International Conference on Shape Modeling and Applications, 2001*, Genova, Italy, May 7-11, pp. 154–166.

Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D. 2002, Shape Distributions, *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 807-832.

Papadakis, P., Pratikakis, I., Perantonis, S. and Theoharis, T. 2007, Efficient 3D Shape Matching and Retrieval Using a Concrete Radialized Spherical Projection Representation, *Pattern Recognition*, vol. 40, no. 9, pp. 2437–2452.

Paquet, E. and Rioux, M. 1999, Nefertiti: A Query by Content System for Three-Dimensional Model and Image Databases Management, *Image and Vision Computing*, vol. 17, no. 2, pp. 157-166.

Paquet, E., Rioux, M., Murching, A., Naveen, T. and Tabatabai, A. 2000, Description of Shape Information for 2-D and 3-D Objects, *Signal Processing: Image Communication*, vol. 16, no. 1, pp. 103–122.

PSMC 2002, *Reduce Program Costs through Parts Management*, Parts Standardization and Management Committee, Defense Standardization Program Office, United States.

Pu, J., Jayanti, S., Hou, S. and Ramani, K. 2006, 3D CAD Model Retrieval Based on Multiple Levels of Detail, *Proceedings of 14th Pacific Conference on Computer Graphics and Applications*, Taipei, Taiwan, October 11-13, pp. 103–112.

Pu, J. and Ramani, K. 2006, On Visual Similarity Based 2d Drawing Retrieval, *Computer-Aided Design*, vol. 38, no. 3, pp. 249–259.

Pu, J., Kalyanaraman, Y., Jayanti, S., Ramani, K. and Pizlo, Z. 2007, Navigation and Discovery in 3D CAD Repositories, *IEEE Computer Graphics and Applications*, vol. 27, no. 4, pp. 38-47.

Rappoport, A., Spitz, S. and Etzion, M. 2005, One-Dimensional Selections for Feature-Based Data Exchange, *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, Cambridge, Massachusetts, United States, pp. 125-134.

Sandor, S. and Leahy, R. 1997, Surface-Based Labeling of Cortical Anatomy Using a Deformable Atlas, *IEEE Transactions on Medical Imaging*, vol. 16, no. 1, pp. 41 - 54.

Saupe, D. and Vranić, D. 2001, 3D Model Retrieval with Spherical Harmonics and Moments, in *Pattern Recognition*, Springer, vol. 2191/2001, pp. 392-397.

Schrijver, A. 2003, *Combinatorial Optimization: Polyhedra and Efficiency. Matroids, Trees, Stable Sets : Chapters 39 -69, Volume 2*, vol. 24, Algorithms and Combinatorics, Springer.

Schröder, B. 2002, *Ordered Sets: An Introduction*, 1 edn, Birkhäuser Boston.

Shah, J. J. and Mäntylä, M. 1995, *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*, John Wiley & Sons, New York.

Shamir, A., Scharf, A. and Cohen-Or, D. 2003, Enhanced Hierarchical Shape Matching for Shape Transformation, *International Journal of Shape Modeling*, vol. 9, no. 2, pp. 203-222.

Shokoufandeh, A., Macrini, D., Dickinson, S., Siddiqi, K. and Zucker, S. W. 2005, Indexing Hierarchical Structures Using Graph Spectra, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1125–1140.

SmarTeam 2006, *Smarteam*, <http://www.3ds.com/products/enovia/>.

Sun, L. and Qamhiyah, A. Z. 2003, Parametric Face Coding for Invariant Model Representation, *Computer-Aided Design*, vol. 35, no. 6, pp. 519–532.

Suzuki, M. T. 2004, A Dynamic Programming Approach to Search Similar Portions of 3D Models, *The World Scientific Engineering Academy and Society Transaction on Systems*, vol. 3, no. 1, pp. 129-135.

Suzuki, M. T., Yaginuma, Y. and Shimizu, Y. 2005, A Partial Shape Matching Technique for 3D Model Retrieval Systems, *The 32nd International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2005*, Los Angeles, California, USA, July 31-August 4.

Suzuki, M. T., Yaginuma, Y., Yamada, T. and Shimizu, Y. 2006, A 3D Model Retrieval Based on Combinations of Partial Shape Descriptors, *Annual Conference of the North American Fuzzy Information Processing Society, NAFIPS 2006*, Montreal, QC, Canada, June, pp. 273-278.

Tam, G. K. L. and Lau, R. W. H. 2007, Deformable Model Retrieval Based on Topological and Geometric Signatures, *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 470-482.

Tangelder, J. W. H. and Veltkamp, R. C. 2008, A Survey of Content Based 3D Shape Retrieval Methods, *Multimedia Tools and Applications*, vol. 39, no. 3, pp. 441-471.

Tarjan, R. 1972, Depth-First Search and Linear Graph Algorithms, *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146-160.

Thomasian, A., Castelli, V. and Li, C.-S. 1998, Clustering and Singular Value Decomposition for Approximate Indexing in High Dimensional Spaces, Bethesda, Maryland, United States, pp. 201-207.

Veltkamp, R. and Latecki, L. 2006, Properties and Performance of Shape Similarity Measures, in *Data Science and Classification*, Springer Berlin Heidelberg, pp. 47-56.

Veltkamp, R. C. and Tanase, M. 2002, *Content-Based Image Retrieval Systems: A Survey*, UU-CS-2000-34, Department of Computing Science, Utrecht University, Utrecht.

Vranic, D. V. 2004, 3D Model Retrieval, Ph.D. thesis, University of Leipzig, Germany.

Vranić, D. V., Saupe, D. and Richter, J. 2001, Tools for 3D-Object Retrieval: Karhunen-Loeve Transform and Spherical Harmonics, *IEEE Fourth Workshop on Multimedia Signal Processing, 2001*, Cannes, France, October 3-5, pp. 293-298.

Wang, D. and Cui, C. 2004, 3D Model Similarity Measurement with Geometric Feature Map Based on Phase-Encoded Range Image, in K. Aizawa, Y. Nakamura and S. Satoh (eds), *Advances in Multimedia Information Processing - Pcm 2004*, Springer-Verlag, Tokyo, Japan, pp. 103-110.

Warshall, S. 1962, A Theorem on Boolean Matrices, *Journal of the ACM*, vol. 9, no. 1, pp. 11-12.

Windchill 2006, *Windchill*, <http://www.ptc.com/products/windchill>.

Xie, J., Heng, P.-A. and Shah, M. 2008, Shape Matching and Modeling Using Skeletal Context, *Pattern Recognition*, vol. 41, no. 5, pp. 1756-1767.

Xie, W., Thompson, R. P. and Perucchio, R. 2003, A Topology-Preserving Parallel 3D Thinning Algorithm for Extracting the Curve Skeleton, *Pattern Recognition*, vol. 36, no. 7, pp. 1529-1544.

Zhang, C. and Chen, T. 2001a, Indexing and Retrieval of 3D Models Aided by Active Learning, paper presented to Proceedings of the ninth ACM international conference on Multimedia, Ottawa, Canada, <http://portal.acm.org/citation.cfm?id=500261>.

Zhang, C. and Chen, T. 2001b, Efficient Feature Extraction for 2d/3D Objects in Mesh Representation, *IEEE International Conference on Image Processing, 2001*, Thessaloniki, Greece, October 7-10, vol. 3, pp. 935–938.

Zhang, J., Siddiqi, K., Macrini, D., Shokoufandeh, A. and Dickinson, S. 2005, Retrieving Articulated 3-D Models Using Medial Surfaces and Their Graph Spectra, in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Springer, vol. LNCS 3757, pp. 285-300.

Zhao, W., Gao, S., Liu, Y. and Lin, H. 2009, Poisson Based Reuse of Freeform Features with Nurbs Representation, *Computers in Industry*, vol. 60, no. 1, pp. 64-74.

Zuckerberger, E., Tal, A. and Shlafman, S. 2002, Polyhedral Surface Decomposition with Applications, *Computers & Graphics*, vol. 26, no. 5, pp. 733-743.