# CAD Parts-Based Assembly Modeling by Probabilistic Reasoning

Kai-Ke Zhang* †, Kai-Mo Hu* †, Li-Cheng Yin* †, Dong-Ming Yan‡ §, Bin Wang* †
*School of Software, Tsinghua University, Beijing 100084, P.R. China
†Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, P.R. China
‡Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia
§National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P.R. China

***Abstract*—Nowadays, increasing amount of parts and subassemblies are publicly available, which can be used directly for product development instead of creating from scratch. In this paper, we propose an interactive design framework for efficient and smart assembly modeling, in order to improve the design efficiency. Our approach is based on a probabilistic reasoning. Given a collection of industrial assemblies, we learn a probabilistic graphical model from the relationships between the parts of assemblies. Then in the modeling stage, this probabilistic model is used to suggest the most likely used parts compatible with the current assembly. Finally, the parts are assembled under certain geometric constraints. We demonstrate the effectiveness of our framework through a variety of assembly models produced by our prototype system.**

***Keywords*-assembly modeling; shape synthesis; probabilistic reasoning; Bayesian network;**

## I. Introduction

Assembly modeling is the process of assembling a set of individual parts for a specific functional goal. The difficulty of the assembly modeling lies in the design of individual parts and setting their constraint information [1], [2]. Since the industrial database stores large numbers of parts and subassemblies, effective reuse of existing models has become a key factor in accelerating the product development cycle and reducing the design cost [3]. To retrieve the desired CAD models from the database, existing methods mainly focus on searching models with geometric, topological and visual similarities [4]. In fact, in the product design stage, people tend to be more concerned about the parts that are most likely to be reused, yet they do not necessarily have visual, topological or geometric similarities with the models in the context of the current assembly.

To gain a global insight based on local observations, the probabilistic models have been introduced to computer graphics for various applications. E.g., the Bayesian network has been used for surface reconstruction [5]–[7]. Many recent work focus on analyzing and generating shape variations from the input shape collections using the machine learning techniques [8]–[11]. Although these approaches are powerful for general modeling purposes, they ignore the aspect that the generated models should satisfy some functional constraints, e.g., mechanical CAD models.

Considering the context of the current assembly and relationships existed in the assemblies, we propose an efficient and intelligent assembly modeling method to facilitate the assembly design process. Our method studies the existing assembly database to learn how parts are assembled together, uses this prior-knowledge to suggest relevant parts to designers at each stage during the modeling process, and assembles these parts together under certain geometric constraints.

We use the Bayesian network [12] to encode the semantic and geometric information between parts in an assembly database. When new parts are assembled, inference in the Bayesian network is used to derive the most relevant parts in the context of the current assembly. For example, when a user begins a modeling task with designing a flange, the system then recommends a pipe fitting or flange washer to the user. The presented parts are dynamically updated as the current subassembly is constructed. To evaluate the presented approach, we have developed a prototype system for assembly modeling. The experimental results demonstrate the effectiveness of our framework.

## II. Related work

Our work is highly related to component based modeling and data-driven modeling. We briefly review the most related approaches in these two areas.

**Component-based modeling.** Funkhouser et al. [1] used a database of segmented shapes to enable interactive assembly of new shapes from retrieved components. Chaudhuri et al. [8] described a data-driven technique for 3D modeling, which computes and presents components that can be added to the artist's current shape. Kalogerakis et al. [9] presented an approach to synthesizing shapes from complex domains, by identifying new plausible combinations of components from existing shapes. Jain et al. [10] described a method that interpolates between two shapes by combining components from these shapes. Chaudhuri et al. [11] developed a probabilistic representation of shape structure that can be used to suggest relevant components during an interactive assembly based modeling session. Furthermore, they extended their framework to semantic attributes based 3D model creatation [13]. These methods mainly focus on the general geometry models, the geometric relationships of which is

978-1-4673-8020-1/15 $31.00 © 2015 IEEE
DOI 10.1109/CADGRAPHICS.2015.29

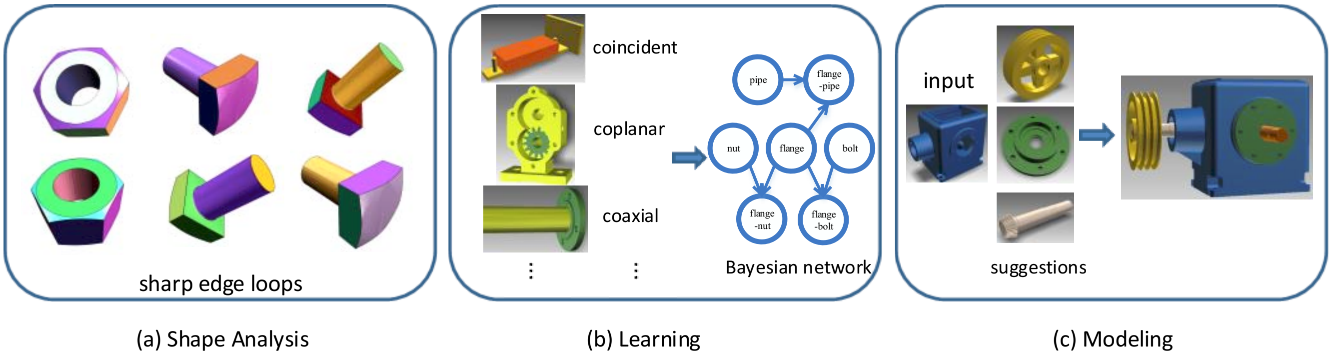



Figure 1: Overview of our approach.

simple and straightforward. While our probabilistic model is substantially different from that of [11] and is designed to tackle CAD models. Unlike the general geometry models, assemblies have more complex geometric constraints and assembly relationships. Thus, the structure of our probabilistic model is more complicated than that of [11].

**Data-driven modeling.** There are numerous representations available for data-driven method, including rule bases, decision trees and artificial neural networks. Fisher and Hanrahan [14] described a method for context-based search of 3D scenes. Afterwards, they proposed how to represent scenes as graphs that encode models and their semantic relationships [15], and presented a method for synthesizing 3D object arrangements from examples [16]. Merrell et at. [17] presented a method for automatic generation of building layouts for computer graphics applications. Given a set of high-level requirements, an architectural program is synthesized using a Bayesian network trained on real-world data. Lin et al. [18] presented a probabilistic factor graph model for automatically coloring 2D patterns. Jain et al. [19] presented a method to assign materials to parts of a 3D object by modelling the context-dependent correlation from a database. Our work applies the data-driven method on assembly modeling, and can present plausible parts to users according to the context of the current sub-assemblies.

The above review shows that the research of general 3D assembly modeling is pretty active. However, we found little work addressing the CAD assembling problem based on data-driven methods. Actually, in mechanical engineering, CAD assembly modeling plays an important role, due to the fact that almost all the products are assemblies rather than single parts [20]. One remarkable difference between general 3D models and CAD assemblies is that in CAD assemblies, the assembly relationships and geometric constraints are far more complicated, for the purpose of assembling design and assembly sequence planning [21]. In this paper, we try to encode these complicated assembly relationships in a graphical model for efficient assembly modeling.

## III. Overview

Our framework consists of three main steps, as illustrated in Figure 1. We first perform shape analysis on the existing parts in the database. Next, a Bayesian network is trained on an assembly database in offline learning stage. Finally, the inference in the probabilistic model is used to present the most relevant parts in the context of the current assembly, and the new parts are assembled according to assembly relationships and geometric parameters.

**Shape analysis.** The purpose of shape analysis is to identify the assembly features and compute the geometric parameters of parts. We follow the work of [22] to extract sharp edge loops from CAD parts (Section IV).

**Learning.** The input of the learning process is an industrial assemblies database. Each assembly is represented as a set of parts (sub-assemblies) and assembly relationships between them. These components are assembled conforming to some assembly guidelines under certain geometric constraints. Our approach involves several assembly relationships defined in commonly used CAD systems, including coincident, coaxial, tangent, coplanar and so on. We employ an assembly tree based method to extract the assembly relationships.

Given the set of parts and their assembly relationships, our method learns a probabilistic graphical model that encodes dependencies between parts as well as their geometric constraint relationships (Section V).

**Modeling.** Finally, we perform the probabilistic inference based on the Bayesian network trained in the learning stage. When new part is added, our probabilistic model suggests the most likely reused parts according to a ranking score. The suggested parts are dynamically updated as the current assembly changes (Section VI).

## IV. Shape analysis

The purpose of shape analysis step is to recognize assembly features of parts, i.e., detecting manufacturing information from models. Examples of this manufacturing information include features such as holes, slots, pockets and other shapes that can be created on modern *Computer Numerically*

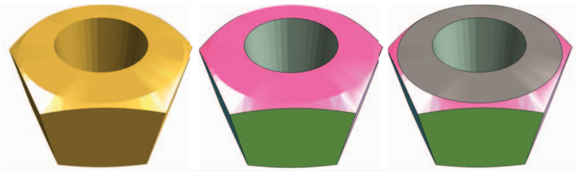
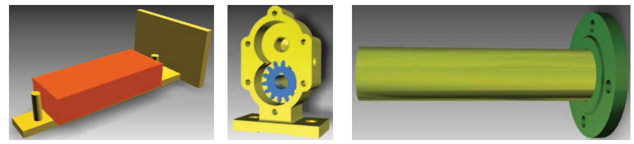


Figure 2: Shape analysis of individual part. Left: input part; middle: sharp edge loops [22]; right: further segmentation by quadric based variational shape approximation [24].



(a) coincident (b) coplanar (c) coaxial

Figure 3: Exemplar relationships of common assemblies. Sometimes two or more relationships may occur in one pair of parts at the same time.

*Controlled* (CNC) machining systems. Three major algorithmic approaches for feature recognition are graph-based algorithms, volumetric decomposition techniques, and hint-based geometric reasoning [23].

In this paper, we analyze the geometric parameters (e.g., translational axis, distance and radius) of each part with the mesh processing methods. The translational axis is used for computation of transformation matrix, and radius is used for filtration of mismatched parts in size. More specifically, we follow the work of Mitra et al. [22] to segment each part into patches by extracting sharp edge loops. For each loop, a simple low degree algebraic surface is fitted to detect the type of the patch [24], [25], and then the planar circular loops can be detected. In general, the circular loops are incident to assembly axes. In this way, we can obtain the geometric parameters (including axes, position, and radius) of parts to assemble. Figure 2 illustrates an example of shape analysis of individual part.

For the models with complex structures that are difficult to analyze, we manually specify their assembly features and corresponding geometric parameters. Using the above method, we obtain the assembly information of each part, which will be used in the assembly synthesis stage later.

## V. Data-driven model learning

The assembly modeling is the process by which the parts and sub-assemblies are assembled together according to assembly relationships and under certain geometric constraints. The semantic and geometric relationships of assembly play an important role during the modeling process.

We use a probabilistic graphical model [12] to learn the structured relationships among parts and geometric constraints between them. To train the probabilistic model, we extract the assembly relationships and record other assembly information of parts for hybrid modeling.

### A. Extracting assembly relationships

Assembly relationships are spatial position relationships that a part must satisfy with respect to other parts [26]. Nettig and Shah [27] gave a comprehensive study about the algebraic constraints, logical constraints, and dimensional constraints. Kim et al. [2] divided assembly relationships into three categories, namely, distance, angle and alignment. These relationships have been used by Mitra et al. [22] to illustrate the motion of the mechanic systems. In general, coincident, perpendicular, tangent, concentric, parallel, distance and angle are the seven most used assembly relationships used by most CAD systems [28]. The purpose of assembly relationships in a design is to control and limit the behavior of the parts in relation to other parts or sub-assemblies as well as for assembly sequence planning.

Different from Chaudhuri's work [11], in which only the adjacency relationship is considered, we define the assembly relationships as listed in Table I. By doing this, the modelling tool will not only suggest the most relative parts, but also their according spatial positions. To give an intuition, we illustrate the according assembly relationships in Figure 3.

Besides the above geometric relationships, some logic relationships also exist between pairs of parts. To extract these relationships, we learn from the assembly library to find pairs of parts appearing with high probabilities in the assemblies. These logic relationships are encoded in our Bayesian network later. Some pairs of the logic relationships that appear in most of assemblies are illustrated in Figure 4.

Table I: Assembly relationship label.

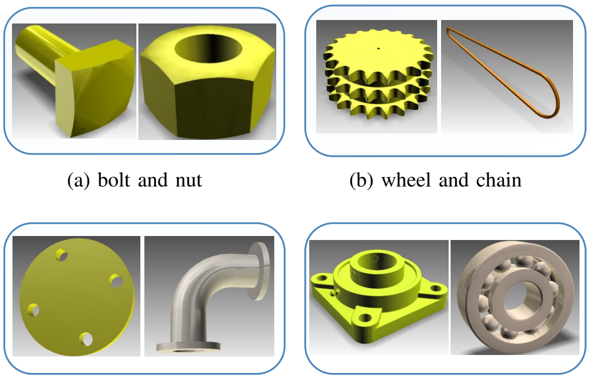
| **Assembly relationship** | **Relationship label** |
|---|---|
| Coincident | C |
| Coplanar | P |
| Coaxial | X |

In our implementation, we employ an assembly tree based method to extract the assembly relationships. In an assembly tree, the whole assembly represents the root of the tree, the sub-assemblies are intermediate nodes, and the parts are leaves of the tree. From the structure of the assembly tree, we can clearly distinguish relationships between the assemblies, sub-assemblies and parts. Given the datasets of assemblies, we traverse each assembly and construct its corresponding assembly tree by inquiring the parts it contained and their relationships. We assign part labels and assembly type labels to parts and assembly relationships, respectively, as illustrated in Table I and Table II.

To this end, an assembly constraint relationship can be represented as $R_{ij}$, in which $R$ signifies assembly relationship shown in Table I, and the numbers $i$, $j$ indicate the types

Table II: Label of some parts used in the paper

| Type | Label | Type | Label | Type | Label |
|---|---|---|---|---|---|
| flange | 1 | screw | 5 | clamp block | 9 |
| bolt | 2 | washer | 6 | chain | 10 |
| wheel | 3 | bearing | 7 | jaw | 11 |
| nut | 4 | bearing block | 8 | pipe fitting | 12 |



(a) bolt and nut

(b) wheel and chain

(c) flange and pipefitting

(d) bearing and bearing block

Figure 4: Exemplar relationships of common assemblies. Sometimes two or more relationships may occur in one pair of parts at the same time.

of parts involved in the relationship. Hence, $R_{ij}$ denotes that, in an assembly component, the types $i$ and $j$ are assembled under relationship $R$. In the assembly process, each step of assembly operation combines two assembly unit (parts or components) together according to certain constraints.

Once the assembly relationships are extracted, we use the graph structure to create semantic association between individual parts. More precisely, we use a Bayesian network to perform relationships modeling, in which the assembly relationships are represented by edges and the parts are represented by nodes.

### *B. Training Bayesian network*

Bayesian network has several advantages: 1) it can easily handle incomplete data sets; 2) it allows one to learn about causal relationships; 3) Bayesian networks in conjunction with Bayesian statistical techniques facilitate the combination of domain knowledge and data; and 4) Bayesian methods in conjunction with Bayesian networks and other types of models offers an efficient and principled approach for avoiding the over fitting of data [29]. Thus, in this paper, we use a Bayesian network to support the data-driven assembly modeling. We first briefly introduce the Bayesian network and then illustrate our modeling process.

A Bayesian network is a graphical model that encodes probabilistic relationships among variables. A Bayesian network for variables $X = \{X_1, ..., X_n\}$ consists of a network structure $S$ that encodes a set of conditional independence assertions about variables in $X$, and a set of $P$ of local probability distributions associated with each variable. The network structure $S$ is a directed acyclic graph. The nodes in $S$ are in one-to-one correspondence with the variables $X$. We use $X_i$ to denote both the variable and its corresponding node, and $\pi(i)$ to denote the parents of node $X_i$ in $S$ as well as the variables corresponding to those parents. Thus, given the structure $S$, the joint probability distribution for $X$ is given by

$$p(x) = \prod_{i=1}^{n} p(x_i|\pi(i)). \tag{1}$$

To train a Bayesian network, we manually encode a large number of industrial assemblies from the database. For each instance, we record the categories of parts, the number of each kind of part and assembly relationships between them. Similar to [11], we represent the attributes as random variables, as illustrated in Table III.

Table III: The random variables used in the model.

| Notation | Range | Interpretation |
|---|---|---|
| $C = \{C_l\}$ | $\{0\} \cup \mathbb{Z}^+$ | part category |
| $N = \{N_l\}$ | $\{0\} \cup \mathbb{Z}^+$ | number of each category |
| $R = \{R_{ll'}\}$ | $\{0, 1\}$ | relationship between $C_l$ and $C_{l'}$ |
| $W = \{W_l\}$ | $R^+$ | weight of category |

For each part category $C_l$, there is a variable $N_l$ that represents the number of parts from $C_l$; for category $C_{l'}$, the variable $R_{l,l'}$ indicates the semantic and geometric relationships between $C_l$ and $C_{l'}$. These random variables help ensure that parts selected for a synthesized model have compatible numbers of parts of each type. For each part category $l$, we also introduce a latent variable $W_l$ that aims to represent the importance of $C_l$ to the current assembly. We assume that the weight of a part is decided by its average size. In our approach, the average size of each part is approximated by the diagonal length of its bounding box. Table IV lists the weights of part categories used in our experiment.

Table IV: The weights of some parts in our experiment.

| Part | Weight | Part | Weight | Part | Weight |
|---|---|---|---|---|---|
| clamp block | 0.96 | jaws | 0.82 | washer | 0.08 |
| flange | 0.50 | pipe | 0.75 | pipe fitting | 0.16 |
| bolt | 0.12 | nut | 0.10 | screw | 0.14 |
| chain | 0.28 | wheel | 0.36 | jaw | 0.42 |

Thus, the Bayesian network represents a joint probability distribution $P(X)$ over all random variables $X = \{C, N, R, W\}$. This distribution is factorized as a product of *Conditional Probability Distributions* (CPDs) as follows:

$$P(X) = P(C_l) \prod_{l' \in \ell} P_{l'},$$
$$P_{l'} = [P(N_l|C_l)P(R_{ll'}|C_l, \pi(R_{ll'}))P(W_l|C_l, \pi(W_l))]. \tag{2}$$

Given a set of training data and a set of variables, our method builds a Bayesian network on these variables. Since the structure and parameters of the network should maximize the posterior probability of the structure of the given data, the algorithm attempts to maximize the Bayesian score

$$\log p(D, S^h) = \log p(D|S^h) + \log p(S^h), \tag{3}$$

where $D$ is the training data and $S^h$ is a model structure. The prior $p(S^h)$ is chosen to be uniform. The marginal likelihood $p(D|S^h)$ is approximated using the Bayesian information criterion

$$\log P(\mathbf{D}|G) \approx \log P(\mathbf{D}|G, \Theta) - \frac{1}{2} v \log(n). \tag{4}$$

Here, $v$ is the number of independent parameters in the network, whereas $n$ is the number of extracted assemblies in the dataset. The first term represents the likelihood of $\Theta$, while the second term keeps the model as simple as possible.

We use the open-source Probabilistic Networks Library [30] to initialize the graphical model. Given the assembly database, we first extract parts and their relationships. There are plenty of special parts that may only appear in a few assemblies, and it is neither reliable nor efficient to encode all the parts in the graphical model. Hence in the learning process, we only consider the relationships between frequently used parts as listed in Table IV. Similar to Chaudhuri' work [11], we also learn the graph structure $G$ and parameters $\Theta$ of the Bayesian network by maximizing the Bayesian Information Criterion score defined in Equation 4.

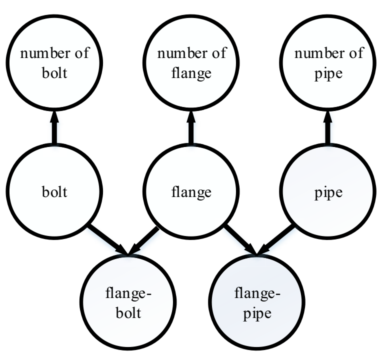



Figure 5: Part of our Bayesian network.

Similar to Merrell et al. [17], we use a local heuristic search that explores the space of network structures by adding, removing and flipping edges. By doing this, we retain the graph structure with the highest score, along with the corresponding parameters for all the CPDs in the model. Figure 5 visualizes a small part of our network generated based on the assembly database.

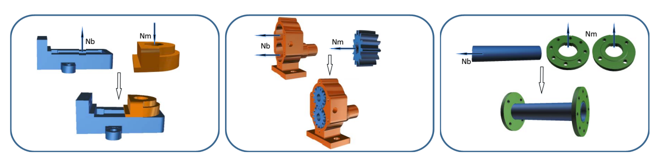
## VI. Assembly synthesis

Once the Bayesian network is trained and the geometric parameters of parts are computed, we can synthesize the assembly model with the help of the trained network. Given the sub-assemblies, the trained network suggests the parts that is most likely be reused according to context of the sub-assembly. The suggested parts are assembled under certain geometric constraints. In this process, we need to specify the relative placement of parts by constraining the relative locations of features on different parts [31].

We follow the work of [32] to transform the well-constrained mating conditions between a base and a mating part into a transformation matrix, which determines the relative orientation and location of the mating part with respect to the base part. In the proposed procedure, users have to first specify the base part, and then the system will compute a transformation matrix, which is determined by a rotation matrix $T_R$ and a translation matrix $T_L$ that define the relative orientation and location of the mating part respectively.

There are three major types of mating conditions: distance, angle, and alignment. The assembly relationships considered in this paper are alignment constraints: coincident, coplanar, and coaxial. More detailed description of these conditions are described below. In the equations below, $N^b$, $N^m$, $N^{mr}$ and $N^{ma}$ indicate the base part, the mating part, the mating part after rotation, and the mating part after assembling, respectively.

**Coincident.** The coincident relationship requires the two faces to touch each other, which is accomplished by constraining the two normal vectors to be opposite to each other, as shown in Figure 6(a).




(a) (b) (c)

Figure 6: (a) Coincident condition, (b) coplanar condition, and (c) coaxial condition.

The face is specified by its unit normal vector $N$ and one point $P$ on the face. Thus, the coincident relationship can be expressed as:

$$N^b = \begin{bmatrix} N_x^b \\ N_y^b \\ N_z^b \end{bmatrix} = - \begin{bmatrix} N_x^{ma} \\ N_y^{ma} \\ N_z^{ma} \end{bmatrix} = -N^{ma}, \tag{5}$$

and

$$\begin{bmatrix} N_x^b \\ N_y^b \\ N_z^b \end{bmatrix} = \begin{bmatrix} P_x^b - P_x^{ma} \\ P_y^b - P_y^{ma} \\ P_z^b - P_z^{ma} \end{bmatrix} = 0. \tag{6}$$

Equations 5 and 6 determine the relative rotation and translation of the mating part with respect to the base part. **Coplanar.** The coplanar relationship requires two planar faces to lie in the same plane, as illustrated in Figure 6(b).

As the coincident condition, the face is specified by its unit normal vector $N$ and one point $P$ on the face. Thus, the coincident relationship can be expressed as:

$$N^b = \begin{bmatrix} N_x^b \\ N_y^b \\ N_z^b \end{bmatrix} = - \begin{bmatrix} N_x^{ma} \\ N_y^{ma} \\ N_z^{ma} \end{bmatrix} = N^{ma}, \tag{7}$$

and

$$\begin{bmatrix} N_x^b \\ N_y^b \\ N_z^b \end{bmatrix} = \begin{bmatrix} P_x^b - P_x^{ma} \\ P_y^b - P_y^{ma} \\ P_z^b - P_z^{ma} \end{bmatrix} = 0. \tag{8}$$

Equations 7 and 8 determine the relative rotation and translation of the mating part with respect to the base part. **Coaxial.** The coaxial relationship holds between two cylindrical faces, which requires the centre axes of shaft and hole part to be coincide, as shown in Figure 6(c). The axis is defined by a unit directional vector and a point on it.

The hole axis is specified by a point $P_b$ and a unit directional vector $N_b$, and the shaft is specified by a point $P_m$ and a unit direction vector $N_m$. Thus, the coaxial relationship can be written as:

$$N^b = \begin{cases} N^{ma}, & \text{aligned coaxial conditions;} \\ -N^{ma}, & \text{opposite-aligned coaxial conditions.} \end{cases} \tag{9}$$

and

$$\frac{P_x^{ma} - P_x^b}{N_x^b} = \frac{P_y^{ma} - P_y^b}{N_y^b} = \frac{P_z^{ma} - P_z^b}{N_z^b}. \tag{10}$$

Equations 9 and 10 determine the relative rotation and translation of the mating part with respect to the base part.

We need to compute the relative orientation and location of the mating part with respect to the base part, which is represented by the transformation matrix. We follow the work of [32] to derive the rotation matrix $T_R$ and translation matrix $T_L$ from the rotational and translational relationships, and the transformation matrix $T$ can be expressed as $T = T_L T_R$.

Note that two parts may have more than one relationship. For instance, the relationship between an axle and a wheel can be coaxial as well as coplanar. When an axle is set as a base part, then a wheel with more than one suggested positions may be recommended by our system. Users can either select the combination of these two relationships, or select one of them, or even discard all of them and specify a new one.

In some cases, there may be more that one possible positions for assembling with respect to the recommended results. To provide a flexible solution for user's assembling process, we rank the possible positions according to the compatibility scores, and set the default position as the one with the highest compatibility score. Users can also manually select their preferred suggested positions or even specify new positions.

Once there is a completely different assembly inserted, our modelling tool can not recommend high-probability parts since there is no existing instance in the training database. In this special case, we return the recommended parts according to their appearing frequency in assemblies. Then users have to select from the database manually or design a new part from scratch. This is the fail case of our modelling tool.

In summary, in the assembly synthesis stage, given the base part and the recommended part, we first filter out the parts with unmatched size, which is computed in the shape analysis process, and then, according to the assembly relationships, we compute the rotation and transformation matrices as described above. Finally, we put the parts in the right position with the right orientation in accordance with the transformation and rotation matrices.

## VII. Experimental results and discussions

In this section, we demonstrate the practical effect of our probabilistic modeling and assembly synthesis method. The assembly database used in our experiments contains 65 real industrial assembly models, which store the information of categories of parts and their assembly relationships. Table V lists the profile of the dataset used in our experiment. The information are collected to carry out the training procedure. Given the fact that some parts appear in very few assembly models, we only encode the parts with a higher frequency of appearance.

Table V: The profile of dataset used in the experiment.

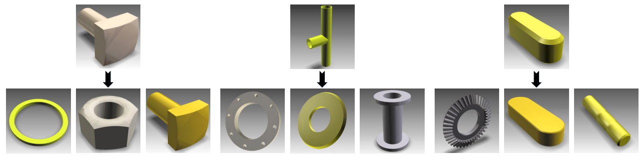
| Name | Count | Name | Count |
|---|---|---|---|
| total assembly models | 65 | categories of parts | 76 |
| total parts in assemblies | 750 | shared parts | 42 |
| average parts per assembly | 11.54 | | |

### A. Recommendation performance

In the modeling stage, the probabilistic graphical model is used to measure the compatibility of each part to the current shape. For each kind of part $l$, we measure the probability of the following formula:

$$\left( \bigvee_{l' \in \ell'} (R_{l_i, l'} = 1) \right) \wedge (N_{l_i} > n_{l_i}), \tag{11}$$

where $l'$ is the type to be measured, $R_{l,l'}$ is the relationship between types $l$ and $l'$, $N_{l_i}$ is the maximum number of part $l'$ that belongs to $R_{l,l'}$, and $n_{l_i}$ is the number of part $l_i$ that



(a) (b) (c)

Figure 7: Recommendation result of individual part. (a) Recommendation result of bolt as base part. (b) Recommendation result of pipe fitting as base part. (c) Recommendation result of key as base part. The top side are three base parts, and the bottom side are the represented parts, respectively.

appear in the current model. The probability of Equation 11 can be computed by $P(X_q|X_e = e)$, in which $X_q \in X$ are random variables, and $X_e \in X$ are observed variables. Thus, the *compatibility score* of part $i$ can be defined as

$$comp(i) = \sum_{q \in Q} P(X_q = q|X_e = e). \tag{12}$$

In each step, the system first computes the compatibility score of each kind of part and recommends parts according to their scores.

In our assembly modeling interface, the recommendation module shows the relevant parts compatible with the base part. Figure 7 shows three examples of recommended results for individual part.

*B. Synthesis results*

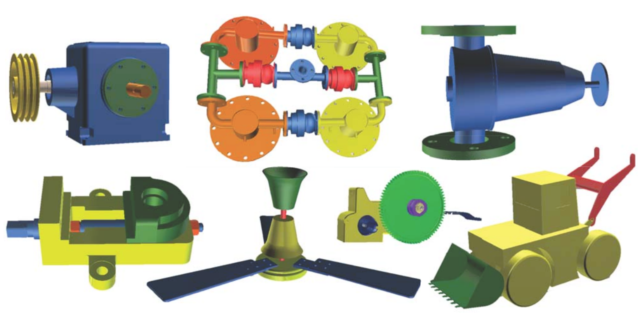


Figure 8: Assembly results created by our modeling tool. The individual parts are shown in different colors.

To evaluate whether our system suggests plausible parts, we have conducted an user study. Six designers who possessed similar proficiency in assembling parts were recruited to assemble the real industry assemblies, as shown in Figure 8. Each of the designers was taught the names and shapes of parts that their tasks required. Then they used the TiGEMS 6.0 as well as our prototype system to assembly the parts. When modeling, our prototype system recommended parts in real time according to the current assembling context in TiGEMS 6.0. The designers could either accept the recommended parts, or deny the recommended parts and retrieve the required parts from the database using existing text or shape based methods.

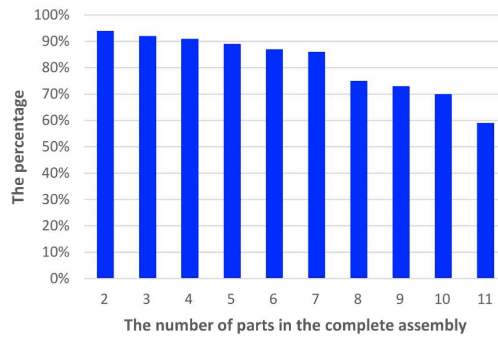



Figure 9: Percentage of parts designers chose from the recommendation list that provided by our prototype system.

Figure 9 presents the average percentage of parts that the designers chose from our recommendation lists. The figure shows that our probability modeling system can recommend most of the right parts for assembling, which improves parts reusability significantly, and thus can improve the efficiency of assembly design. Note that when the assemblies get more complicated, the recommendation effect becomes worse. This results from the facet our assembly library is limited. If there is no existing instance in the training database that are similar to the current assembling context, then designers have to search parts from the database manually.

## VIII. Conclusion and Future Work

This paper proposes an efficient and intelligent assembly modeling framework to improve design efficiency of CAD assemblies. We successfully employ the data-driven method in CAD assembly modeling. Given a repository of assembly models, an assembly tree based method is used to extract the assembly relationships between parts. Then, a probabilistic graphical model is learned for relationship modeling. More specifically, our approach learns a Bayesian network that encodes the assembly relationships among parts in the preprocessing stage. In the modeling stage, inference in the Bayesian network is used to suggest relevant parts to designers. Experiments verified its effectiveness.

One limitation of our work is that the assembly synthesis method can only handle parts with uncomplicated assembly relationships and less geometric constraints. To synthesize more complex models, we shall study more powerful algorithm to solve assembly synthesis problem.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (61373071, 61372168).

## REFERENCES


[1] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin, "Modeling by example," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 652–663, 2004.

[2] J. Kim, K. Kim, J. Y. Lee, and H. B. Jung, "Solving 3D geometric constraints for closed-loop assemblies," *International Journal of Advanced Manufacturing Technology*, vol. 23, pp. 755–761, 2004.

[3] T. G. Gunn, "The mechanization of design and manufacturing," pp. 114–130, 1982.

[4] K. Hu, B. Wang, B. Yuan, and J. Yong, "Automatic generation of canonical views for CAD models," in *CAD/Graphics 2011*, 2011, pp. 17–24.

[5] Q.-X. Huang, B. Adams, and M. Wand, "Bayesian surface reconstruction via iterative scan alignment to an optimized prototype," in *Proc. Symp. on Geometry Processing*, 2007.

[6] P. Jenke, M. Bokeloh, A. Schilling, W. Straßer, and M. Wand, "Bayesian point cloud reconstruction," in *EUROGRAPHICS 2006*, 2006.

[7] J. R. Diebel, S. Thrun, and M. Brünig, "A bayesian method for probable surface reconstruction and decimation," *ACM Trans. Graph.*, vol. 25, no. 1, pp. 39–59, Jan. 2006.

[8] S. Chaudhuri and V. Koltun, "Data-driven suggestions for creativity support in 3d modeling," in *ACM SIGGRAPH Asia 2010 Papers*, ser. SIGGRAPH ASIA '10. ACM, 2010, pp. 183:1–183:10.

[9] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun, "A probabilistic model for component-based shape synthesis," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 55:1–55:11, Jul. 2012.

[10] A. Jain, T. Thormählen, T. Ritschel, and H.-P. Seidel, "Exploring shape variations by 3d-model decomposition and part-based recombination," *Comp. Graph. Forum*, vol. 31, no. 2pt3, pp. 631–640, 2012.

[11] S. Chaudhuri, E. Kalogerakis, L. Guibas, and V. Koltun, "Probabilistic reasoning for assembly-based 3D modeling," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 30.

[12] D. Koller and N. Friedman, "Probabilistic graphical models: Principles and techniques," *The MIT Press*, 2009.

[13] S. Chaudhuri, E. Kalogerakis, S. Giguere, and T. Funkhouser, "AttribIt: Content creation with semantic attributes," in *Proc. UIST*. ACM, 2013.

[14] M. Fisher and P. Hanrahan, "Context-based search for 3d models," in *ACM SIGGRAPH Asia 2010 Papers*, 2010, pp. 182:1–182:10.

[15] M. Fisher, M. Savva, and P. Hanrahan, "Characterizing structural relationships in scenes using graph kernels," in *ACM SIGGRAPH 2011 Papers*, 2011, pp. 34:1–34:12.

[16] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan, "Example-based synthesis of 3d object arrangements," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 135:1–135:11, 2012.

[17] P. Merrell, E. Schkufza, and V. Koltun, "Computer-generated residential building layouts," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 181:1–181:12, 2010.

[18] S. Lin, D. Ritchie, M. Fisher, and P. Hanrahan, "Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 37:1–37:12, Jul. 2013.

[19] A. Jain, T. Thormählen, T. Ritschel, and H.-P. Seidel, "Material memex: Automatic material suggestions for 3d objects," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 143:1–143:8, 2012.

[20] K.-M. Hu, B. Wang, J.-H. Yong, and J.-C. Paul, "Relaxed lightweight assembly retrieval using vector space model," *Computer-Aided Design*, vol. 45, no. 3, pp. 739–750, 2013.

[21] K.-M. Hu, B. Wang, Y. Liu, J. Huang, and J.-H. Yong, "An extended schema and its production rule-based algorithms for assembly data exchange using iges," *The International Journal of Advanced Manufacturing Technology*, vol. 58, no. 9-12, pp. 1155–1170, 2012.

[22] N. J. Mitra, Y.-L. Yang, D.-M. Yan, W. Li, and M. Agrawala, "Illustrating how mechanical assemblies work," *Commun. ACM*, vol. 56, no. 1, pp. 106–114, 2013.

[23] J.-H. Han, M. Pratt, and W. Regli, "Manufacturing feature recognition from solid models: A status report," 2000.

[24] D. Yan, Y. Liu, and W. Wang, "Quadric surface extraction by variational shape approximation," in *Geometric Modeling and Processing - GMP 2006*, 2006, pp. 73–86.

[25] D. Yan, W. Wang, Y. Liu, and Z. Yang, "Variational mesh segmentation via quadric surface fitting," *Computer-Aided Design*, vol. 44, no. 11, pp. 1072–1082, 2012.

[26] J. Kim, M. J. Pratt, R. G. Iyer, and R. D. Sriram, "Standardized data exchange of cad models with design intent," *Computer-Aided Design*, vol. 40, no. 7, pp. 760–777, 2008.

[27] B. Bettig and J. Shah, "Derivation of a standard set of geometric constraints for parametric modeling and data exchange," *Comput.-Aided Des.*, vol. 33, no. 1, pp. 17–33, 2001.

[28] *Mastering Cad/Cam (Sie)*. McGraw-Hill Education (India) Pvt Limited.

[29] D. Heckerman, "Learning in graphical models," M. I. Jordan, Ed. Cambridge, MA, USA: MIT Press, 1999, ch. A Tutorial on Learning with Bayesian Networks, pp. 301–354.

[30] V. Eruhimov, K. Murphy, and G. Bradski, "Intel's open-source probabilistic networks library."

[31] R. Anantha, G. A. Kramer, and R. H. Crawford, "Assembly modeling by geometric constraint satisfaction," *Computer Aided Design*, vol. 28, no. 9, pp. 707–722, 1996.

[32] J. Kim, K. Kim, K. Choi, and J. Lee, "Solving 3d geometric constraints for assembly modelling," *The International Journal of Advanced Manufacturing Technology*, vol. 16, no. 11, pp. 843–849.