Graphical Models 92 (2017) 1–15

Contents lists available at ScienceDirect

Graphical Models

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

Topology based 2D engineering drawing and 3D model matching for process plant

Rui Wen[a,*], Weiqing Tang[a,b,d], Zhiyong Su[c]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
[b] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[c] School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China
[d] Beijing Zhongke Fulong Computer Technology Co., Ltd, Beijing 100085, China

ARTICLE INFO

Article history:
Received 20 September 2016
Revised 15 May 2017
Accepted 10 June 2017
Available online 12 June 2017

Keywords:
Process plant
3D model
2D engineering drawing
Feature similarity
Matching calculation

ABSTRACT

Process plant models mainly include 3D models and 2D engineering drawings. Matching calculation between these CAD models has wide applicability in model consistency check and retrieval. In process plant, engineering design standards make 2D engineering drawing and 3D model differ in geometry, proportion and structure, leading to the inapplicability of current shape-feature based matching approaches. Since connection relationships between components are the core of a process plant, a topology based algorithm is proposed. Firstly, by exploiting components as vertices and relationships as edges, both 2D engineering drawing and 3D model are preprocessed into graph structures. Then each model's relationship types are extracted from the graph. Finally, regarding the extracted relationship types as primary feature, feature similarity is calculated to measure the matching degree between their corresponding models. The proposed algorithm is geometric deformation invariant. Experiments with industrial applications are presented, which demonstrates the effectiveness and feasibility of the proposed algorithm.

1. Introduction

2D engineering drawings and 3D models (2D & 3D) are crucial documents in plant design and construction. In a timely fashion, a process plant's life cycle can be divided into planning, design, construction, operation and disposal [1]. Thereinto, design stands out as the most important stage in ensuring the plant quality. In this stage, designers should first construct a high-quality and integrated 3D model according to the achievements of planning stage, as shown in Fig. 1. Then based on this 3D model, different types of 2D engineering drawings are made as clear and precise guidance for construction stage, as shown in Fig. 2.

An effective 2D & 3D matching calculation is imperative to improve design quality and efficiency. The matching result not only contributes to model consistency check, but also benefits for model retrieval.

(1) With the increasing scale of process plant, the traditional man-powered consistency check is becoming much more time and effort consuming. Design consistency must be verified before construction in order to assure plant quality. Therefore, designers have to manually go through every detail of components and relationships. Comparatively, an automatic 2D & 3D matching calculation can help designers to improve consistency check efficiency.

(2) The huge number of CAD models brings out the low efficiency in model management. Take the large Chinese nuclear power project – Qinshan Nuclear Power Plant Project for example, the number of CAD models is already up to 40,000 only in its second phase [2]. Due to the difficulty of model text acquisition and lack of standardization, a filename based model retrieval can no longer meet enterprise-level needs. Comparatively, content-based model retrieval serves as a useful searching method through exploiting a model's internal characteristics. As the key point of model retrieval, an effective matching calculation is necessary.

Consequently, given a 2D engineering drawing and a 3D model, our goal is to automatically calculate their matching degree.

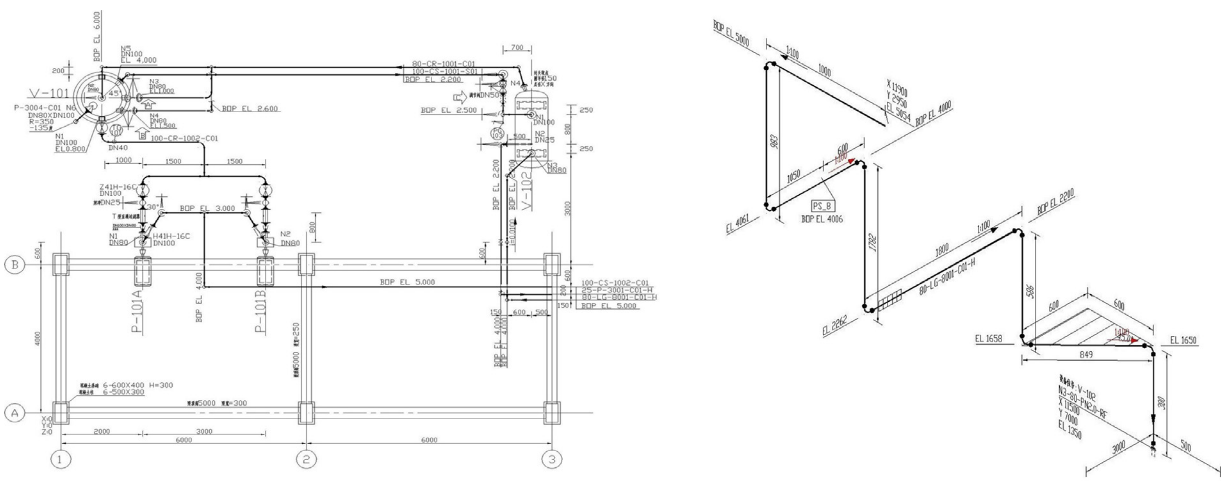
Due to the engineering design standards, process plant 2D & 3D matching calculation faces the following difficulties:

(1) **Geometry.** A component can have various 2D geometries which are irrelevant to its 3D shape projection. Graphics library provides different graphical symbols for each component according to the drawing type and engineering

* Corresponding author.
E-mail address: wenruity@163.com (R. Wen).

http://dx.doi.org/10.1016/j.gmod.2017.06.001
1524-0703/

Fig. 1. A hydrocarbon plant model.



(a) Pipeline plane drawing. (b) ISO drawing.

Fig. 2. 2D engineering drawings.

attribute. That is to say, components with the same type may be represented by different geometries. As a result, it is difficult to build an exact correspondence between a component's 2D geometry and 3D shape.

(2) **Proportion.** Geometric deformations may occur in non-scale 2D engineering drawings. Based on the usages of 2D engineering drawings, some types of drawings (such as ISO drawings) are disproportionate with their corresponding 3D models. In order to improve the aesthetics and readability of these drawings, pipelines should be properly zoomed in or out according to the area's density. Thus, different proportion scales may lead to the significant differences between 2D engineering drawings and their 3D models.

(3) **Structure.** The content structures that 2D engineering drawings described could be global or partial structures of 3D models. The uncertainty of content structures increases the calculation complexity of this proposed problem.

In conclusion, there still exists geometry, proportion and structure changes between 2D engineering drawings and 3D models, even if they are derived from the same plant. Therefore, a successful 2D & 3D matching method should handle the difficulties above.

In most cases, people prefer to differentiate models from their appearances. Intuitively, lots of researchers tend to measure model similarity from a shape perspective [3–8]. According to this, existing approaches calculate geometric similarity between 2D drawing and 3D model's projection drawing to measure 2D & 3D matching degree [9–14]. However, process plant models have variances in geometry, proportion and structure. These variances lead to the significant geometric differences between 2D engineering drawing and 3D model's projection drawing, thus making existing approaches inapplicable.

Prior to this work, we have presented two researches on process plant model similarity. The first research is a 3D model similarity algorithm [15]. This algorithm is a global similarity study within only 3D models. It aims to retrieve top-n similar models from existing model database, so as to ensure new model quality and guide future design. This algorithm constructs a tree structure to extract the distribution of a model's relationships and measures the model similarity according to their relationship distributions. The second research is a 2D & 3D matching algorithm [16]. This algorithm transforms a 2D drawing and a 3D model into graph structures and calculates their Min-edit Distance to measure the 2D & 3D matching degree. However, this algorithm presents some disadvantages in efficiency and stability, because insertion coordinate is the only feasible attribute allowed by the algorithm.

The first contribution of this paper is the design of a framework that decomposes the proposed matching algorithm into three parts: attribute graph construction, feature extraction and matching calculation.

- Attribute graph construction is fundamental of this framework. Considering the aim of process plant is to construct the plant layout [17], the topology structures of a 3D model and its 2D engineering drawings should remain invariant. The geometric changes brought by proportion and component's geometry could be solved if the measure point lies on topology structure. Intuitively, the 2D & 3D matching can take topology structure as a starting point. Graph-based model similarity has a good performance in measuring topology similarity [18–24]. Thus, we seek to transform CAD models into attribute graphs and evaluate 2D & 3D matching degree through their graph similarity. In real engineering applications, 2D engineering drawings and 3D models are stored as parameterized structures in CAD documents [25]. We can make use of a CAD document to construct a model's attribute graph [26].
- Feature extraction is applied to measure graph similarity after constructing attribute graphs. Common solutions of graph similarity are Graph Edit Distance [27–29] and Maximum Common Sub-graph [30]. However, their efficiency faces enormous challenge because computing the edit distance is NP-hard and maximal common sub-graph is also NP-complete. Therefore, feature based solution is used in this paper. By applying the same feature extraction method to both graphs, each graph is represented by a set of features [31]. At this point, the problem of graph similarity can be boiled down to feature similarity.
- Matching calculation is imperative to compare the feature similarity. As a 2D engineering drawing can be a global or partial structure of its 3D model, an effective method should handle these situations.

The second contribution of this paper is the new feature extraction method. In this method, type attribute is regarded as a component's primary identification, because designers usually differentiate components according to their types; connection relationship types are used as a model's feature, as CAD models' relationships will be invariant as long as they belong to the same process plant. The extraction method acquire a model's feature by traversing each model's relationships.

The third contribution is the matching calculation method. First, the criteria for determining whether a 2D engineering drawing and a 3D model matches are defined. Remember that a 3D model and its 2D drawings could differ in geometry, proportion and structure, these criteria should be able to handle all these situations. Then, based on each model's topology feature and the criteria defined above, a matching calculation approach is presented to achieve the similarity assessment. In this approach, the model with less relationship types is chosen as reference. According to the reference model's relationship types, the compared models' unique feature vectors can be acquired. Finally, 2D & 3D matching degree can be calculated through the cosine value of feature vectors. Thus, the problem caused by different content structures can be efficiently bypassed.

The rest of this paper is organized as follows: domain characteristics of 3D model and 2D engineering drawing in process plant are introduced in Section 2, with explanations of problem identification. Section 3 reviews related work in 2D & 3D matching research, graph-based model similarity research and our previous work. Section 4 gives an in-depth description of the proposed framework, followed by experiments and discussions in Section 5. In Section 6, we summarize the whole paper with conclusions and future work.

2. Process plant models and problem identification

Process plant is the set of reaction vessels, pipelines and supports which are materials for making chemical or physical manufactured products [32]. In this section, we will first give a brief introduction to its 3D model and 2D engineering drawing.

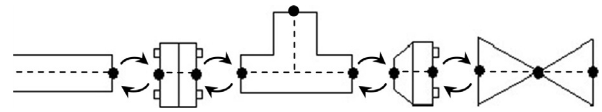


Fig. 3. A simple duality point and smart line based pipeline example.

2.1. 3D models

Process plant 3D models should precisely describe every connection relationship between components, as well as some design constraints. The ultimate goal of constructing 3D models is to generate different types of 2D engineering drawings for constructional guidance.

2.1.1. Component

Different from other grid or surface constituted models, a quintessential plant model consists of hundreds of thousands of components. These components include equipments, pipelines (i.e. pipes and piping components), valves, instruments, etc. In more detail, components in process plant are comprised of fourteen basic entities: cylinder, scylinder, prism, econe, concone, squcir, squcone, box, torus, squtorus, sphere, wedge, saddle and oval.

2.1.2. Connection relationship

Connection relationship between components plays a crucial role in the accurate expression of topology structures. As illustrated in Fig. 3, relationships are represented by the dual point and smart line based method [33]. In this method, dual points (the black dots in Fig. 3) attempt to describe component-wise topologies. Two components' dual points will be coincident on the joint only if they have a connection relationship. Smart lines (the dotted lines in Fig. 3) seek to build up the relational constraints between pipes and pipelines by abstracting them as 3D segments and curves.

2.1.3. Engineering attribute

Engineering attribute is essential to the precise descriptions of components and relationships. It usually contains information about design constraints, engineering disciplines, etc. For example, type attribute stands out as a component's unique identifier in engineering databases; flow direction records a relationship's liquid direction; insertion coordinate, which can be considered as the reference coordinate of a relationship, is the coordinate of dual points (the black dots in Fig. 3) in 3D space. Other attributes, like material, pipeline level, facing type, wall thickness, etc., fully describe the plant and make it better understandable for builders.

2.2. 2D engineering drawings

2D engineering drawings exploit 2D graphs as well as annotations and illustrations to describe the details of a 3D model [2]. For the status quo of engineering CAD development, designs still largely remain in the form of 2D engineering drawings.

To provide clear guidance for construction, 2D engineering drawings are divided into various types based on the main usage, such as piping layout drawing, equipment layout drawing, connection orientation drawing, ISO drawing. Meanwhile, to improve the aesthetics and readability, engineering design standards are established by related industry associations, leading to the particular properties of 2D engineering drawings.

First, a component can have various 2D geometric descriptions. Graphics library is usually introduced to define graphical symbols

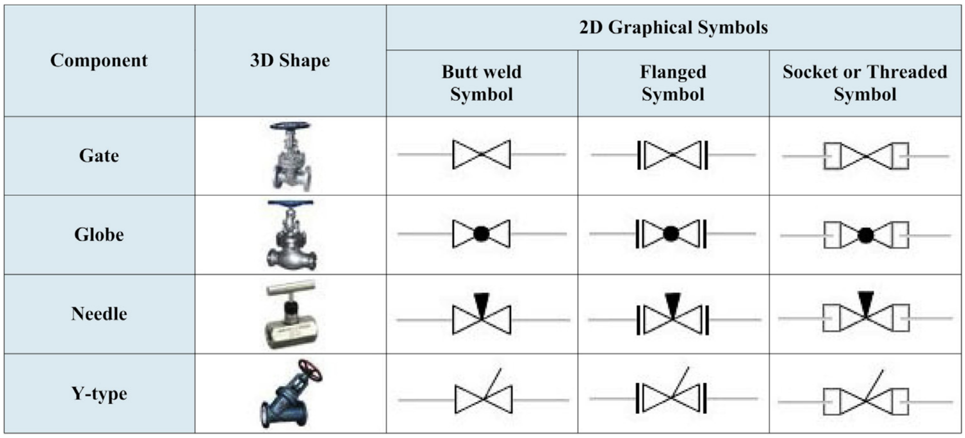
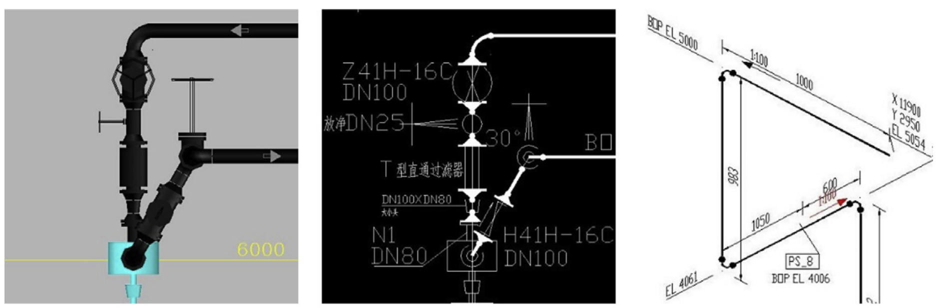


Fig. 4. Different graphical symbols of components.



(a) Projection drawing. (b) Plane drawing. (c) ISO drawing.

Fig. 5. The zoomed-in views of different drawings. Drawings in (a), (b) and (c) come from the same plant. (a) and (b) describe the same location of the plant.

for each component type. As Fig. 4 illustrates, different connection modes (such as butt weld and flanged connections) are assigned with corresponding symbols. In other words, components with the same type may be represented by different graphical symbols.

Second, a plant's 2D engineering drawings may vary with different projection methods. Besides the well-known proportional projection, some types of drawings are non-scale projections of 3D models. Take the ISO drawing illustrated in Fig. 2(b) for example, this type of drawings is the isometric projection of pipe systems. For guiding purpose, complex relationships and related annotations should be clearly described in a limited plane. Consequently, practical construction and engineering design standards allow ISO drawing to be non-scale and components to be properly deformed.

Last, a plant's 2D engineering drawings could differ in content structure. Fig. 2 gives two instances which are derived from the same 3D model, as are a pipeline plane drawing and an ISO drawing. The pipeline plane drawing describes a plant's global structure and the ISO drawing describes a partial structure of interest.

2.3. Problem identification

The variances in geometry, proportion and structure make process plant 2D & 3D matching difficult. Geometric differences still exist between 2D engineering drawing and 3D model's projection drawing even if they come from the same plant. These differences lead to the inapplicability of existing 2D & 3D matching approaches.

First of all, a component's 2D geometry is not the direct projection of its real 3D shape, but the graphical symbols prescribed by graphics library. For example, Fig. 5(a) and (b) are the zoomed-in views of a 3D model's projection drawing and plane drawing respectively. Obviously, the geometric descriptions of components in Fig. 5(b) have differences with their 3D shape projections in Fig. 5(a).

Then, the non-scale projection leads to the geometric deformations of components. Some specified types of drawings aim to depicting the relative relations (i.e., topology) between pipes. In order to improve their aesthetics and readability, engineering design standards permit some necessary and proper deformations. Take Fig. 5(c) for example, the components in this drawing have been significantly deformed even though it is a drawing of 3D model in Fig. 5(a).

Moreover, differences in content structures further add to the complexity of 2D & 3D matching problem. Some types of drawings describe a global structure to overview a plant's layout. Some types of drawings describe a plant's partial structure to introduce the local detail.

3. Related work

There have been several branches in 2D & 3D matching research, mostly regarding general models and product CAD models. We summarize some accepted typical methodologies in this section.

3.1. 2D & 3D Matching for general models

General models are models that human can make contact with in daily life, such as furniture models, animal models, etc. Their 2D & 3D matching research focuses largely on geometric feature measures, as they are all built-up with triangular meshes [18].

Wang et al. [9] present a view-based discriminative probabilistic modeling for 3D object retrieval. Based on the distribution of views, they first build probabilistic models for each object; secondly, the distance between two objects is defined as the upper bound of the Kullback–Leibler divergence of the corresponding probabilistic models; lastly, 3D object retrieval and recognition is accomplished according to the distance calculations.

Min et al. [10] propose a single depth image based 3D model retrieval method. In their effort, 3D models are represented by multiple depth images acquired from adaptively sampled viewpoints. A single depth image, which can be easily captured with an off-the-shelf low-cost 3D camera, is used as the input query. Their proposed algorithm can retrieve relevant 3D models while considering local 3D geometric characteristics using a rotation-invariant feature descriptor.

Lahner et al. [11] present a non-rigid 2D-to-3D shape matching algorithm. This algorithm's input are a 2D shape represented as a planar curve and a 3D shape represented as a surface. The output is a continuous curve on the surface. They solve the problem by finding the shortest circular path on the product 3-manifold of the surface and the curve.

The techniques above represent 3D models by probabilistic models or images from different viewpoints and measure the shape similarity from a geometric perspective. In the meantime, some other researches are inspired by model's topology. For example, Waleed et al. [12] introduce a skeletonization algorithm which utilizes a normalized mixture distance function to encode a 3D shape into a topological Reeb graph. In addition, they also propose a novel graph matching algorithm by comparing the relative shortest paths between the skeleton endpoints.

3.2. 2D & 3D Matching for product CAD models

Product CAD models focus on the geometric descriptions of mechanical products, electrical and electronic products. Because product CAD models are mainly comprised of curves and surfaces, their retrieval system concentrates on the research of surface feature computation, free-form surface parameterization and mechanical specific standardization.

Li et al. [13] retrieve 3D product CAD models using 2D images with optimized weights. First, the Pyramid Histogram of Oriented Gradients (PHOG) descriptor is employed to describe the 2D images projected from a model. Then, Lagrange multipliers, vector quantization and a Support Vector Machine (SVM) are used to adaptively assign an optimal weight to each projected image.

Pu et al. [14] present a sketch user interface enhanced by feedback for 3D product CAD model retrieval. Through this method, users can not only emphasize some shapes by specifying weights for views, but also do some editing operation on the views obtained from the retrieved models. Thus, users can express their intent by sketching 2D shape in the way as engineers draw three views of 3D models. The whole retrieval process forms a loop by which users can refine the search results step by step.

Based on subgraph isomorphism, Huang et al. [34] introduce a matching algorithm between precursory 3D process model and 2D working procedure drawing. Firstly, they obtain the projection drawing of the precursory 3D process model; then extract the primitives and construct the attributed adjacency graph; finally, by taking the 2D working procedure drawing and projection drawing as the attributed adjacency graphs, the matching problem between precursory 3D process model and 2D working procedure drawing is translated into the problem of subgraph isomorphism.

In conclusion, the 2D & 3D matching approaches above pay most of their attentions to shape-feature similarity. Basically, these researches can be roughly summarized into three steps [35]:

(1) acquire the 3D model's projection drawing from the same viewpoint with the compared 2D drawing;
(2) apply shape feature extraction to both drawings;
(3) calculate shape feature similarity to measure the matching degree between the compared 2D drawing and 3D model.

Noticeably, these approaches have a strong assumption on step (1). However, engineering design standards may change the component's 2D appearance, making it irrelevant from its 3D shape projection. Therefore, these shape-feature based approaches cannot solve the proposed problem.

3.3. Graph-based model similarity

Graph-based similarity is extensively used in model retrieval area due to its inherent ability in describing model topology. Algorithms of this category are dedicated to transforming the research target into a graph structure and seek to measure model similarity through their corresponding graphs.

Sundar et al. [19] propose a skeleton based method for comparing 3D objects. This method encodes the geometric and topological information in the form of a skeletal graph and uses graph matching techniques to match the skeletons and to compare them. First, they compute the skeletal graph from the volumetric object. Then, they store along with the graph, the topological signature vector of the graph, which is a low-dimensional index that captures both the local and global structural properties. Last, the object similarity can be calculated according to their topological signature vectors.

Ayellet et al. [20] present a mesh retrieval method by components. Their key idea is to represent an object by an attributed graph that consists of the objects meaningful components as nodes, where each node is fit to a basic shape. Accordingly, given a database of meshes in a standard representation consisting of vertices, faces and one specific object *O*, the goal is to retrieve from the database objects similar to *O*. To retrieve the similar objects, they calculate the edit distance between attributed graphs to measure the corresponding object similarity.

Demirci et al. [21] propose an approach for indexing multimedia databases in which entries can be represented as graph structures. In this study, the topological structure of a graph as well as that of its subgraphs are represented as vectors whose components correspond to the sorted laplacian eigenvalues of the graph or subgraphs. Since the laplacian spectrum is used as a graph signature, a high level of uniqueness is maintained. Therefore, they compare these signatures of a large number of graphs without solving the computationally expensive correspondence problem between their vertices.

By using both topological and geometric features at the same time, Gary et al. [22] propose Topological Point Ring (TPR) analysis to locate reliable topological points and rings firstly. Then, they capture both local and global geometric information to characterize each of these topological features. To compare the similarity of two models, they adapt the Earth Mover Distance (EMD) as the distance function and construct an indexing tree to accelerate the retrieval process.

Shapira et al. [23] present a framework which automatically finds part analogies among 3D objects. This method first partitions a given 3D object to create a part hierarchy, and then defines a signature for each part. Using these signatures they define an effective context-aware distance measure that can find analogous parts among other objects, which are not necessarily similar as a whole.

Kleiman [24] present shape edit distance (SHED) that measures the amount of effort needed to transform one shape into the other. The shape edit distance takes into account both the similarity of the overall shape structure and the similarity of individual parts of the shapes.

Obviously, the algorithms above also emphasize on shape-feature similarity. Their ideas can be summarized as follows:

(1) decompose a model into substructures and represent these substructures by a graph's vertices;
(2) extract geometric or topological features of the substructures and regard the features as the vertices' attributes;
(3) calculate attribute graph similarity to measure the corresponding model similarity.

However, these algorithms are inapplicable in process plant 2D & 3D matching calculation. On one hand, it is difficult to build an actual correspondence between a component's 2D and 3D shapes. A component can have various graphical symbols as introduced in Section 2.2. That is to say, in a 2D engineering drawing, components with the same type may be represented by different shapes. On the other hand, a 2D engineering drawing and its 3D model's projection drawing can differ in appearance and content structure, making it hard to compare 2D & 3D shape similarity. Even so, the idea that transforming a model into an attribute graph can still make contributions to solving the proposed problem.

3.4. Our previous work

Prior to this work, several process plant model similarity researches have been conducted by our team. A similarity measurement centered in homogeneous 3D model is proposed in [15]. This algorithm can help to find the intermediate 3D models generated during the collaborative design process. The retrieved models are rewarding for the following procedure to avoid unnecessarily repeated design. In this work, a tree model for extracting relationship distribution is first proposed. Second, standardization is performed via mapping relationship statistics into vector space to achieve the comparable feature vector. Last, a hybrid similarity function combining both directional and numerical differences in feature vectors is proposed to evaluate 3D model similarity.

Besides 3D model similarity, heterogeneous 2D & 3D matching is also important in process plant as explained in Section 1. Their global or partial similarity can help to determine whether they come from the same process plant, so as to verify model coherency and enhance management efficiency. To tackle this problem, a 2D & 3D matching algorithm using Min-Edit Distance is presented in [16]. However, it presents some disadvantages in efficiency and stability. Here we give a brief discussion.

Min-Edit Distance [27,28] is a common solution of graph similarity. In general, graph similarity can be roughly divided into precise measurement and error-tolerant measurement [29], each with a representative solution of Max-Common Sub-graph and Min-Edit Distance. Bunke puts forward in [36] that given a certain cost function, Min-Edit Distance will become equivalent to Max-Common Sub-graph [30]. As Edit Distance is defined within a common theoretical framework[1], it is actually a quite flexible method and has been successfully applied to model similarity field [20,24,37]. So as a natural choice, this algorithm seeks to compare 2D & 3D matching degree based on Min-Edit Distance by transforming CAD models into attribute graphs.

One major disadvantage of this Min-Edit Distance based algorithm is: within various attributes in a relationship, insertion coordinate is the only feasible attribute allowed by the algorithm. Different relationships' insertion coordinates can hardly coincide in a 3D space, hence the correspondent relationships are easy to find out. However, once coordinate information go awry (e.g., rotation, translation), or in some cases it is simply not the deciding measure emphasis, this algorithm would fail.

[1] The min-edit distance between graph G_1 and G_2 is the minimum number of operations to transform G_1 into G_1' such that $G_1' = G_2$. The primitive edit operations on a graph include insert an isolated vertex with label, delete an isolated vertex, substitute a vertex label, insert an edge between two vertices, delete an edge and substitute an edge label.

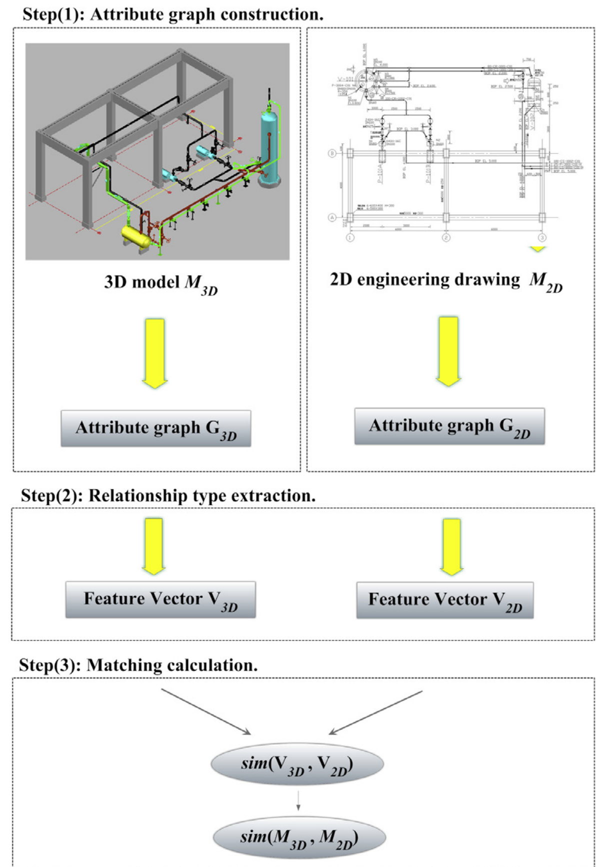


Fig. 6. Algorithm diagram.

Koutra concludes another solution to compare graph similarity in [31], that is applying the same feature extraction method to both graphs, and feature based difference is then used against graph similarity. In this solution, feature extraction and similarity calculation are the major challenges. Inspired by this solution, we can commence by extracting a model's topology feature, followed by calculating feature similarity to measure the matching degree between their corresponding models.

4. Algorithm detail

4.1. Overview

Process plant has a strong emphasis on the accurate topology expression between components. CAD models' topology structures will be consistent as long as they belong to the same process plant, even though they may differ in lots of aspects (as mentioned in Section 2.3). More importantly, this rule also applies even in different dimensional representations. Intuitively, we can measure 2D & 3D matching degree from a topology perspective.

By representing components as vertices and relationships as edges, a plant model can be transformed into a topological structure, which is indeed an attribute graph. In the meantime, relationship types can be further regarded as a model's topology feature. So a topology based 2D & 3D matching algorithm is proposed in this paper. As illustrated in Fig. 6, this algorithm has three primary steps:

(1) attribute graph construction;
(2) relationship type extraction;
(3) matching calculation.

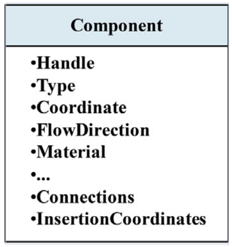


Fig. 7. The parameterized structure of a component.

In the first step, geometric variations brought by proportion and components' geometries can be bypassed through transforming CAD models into attribute graphs. In the second step, the problem of graph similarity falls into feature similarity by regarding extracted relationship types as a model's topology feature. In the last step, the problem caused by different structures can be resolved by using the model with less relationship types as reference.

4.2. Attribute graph construction

In real engineering applications, 2D engineering drawings and 3D models are stored as parameterized structures in CAD documents. Without loss of generality, a component can be described as Fig. 7 shows:

- **Handle** is a dynamically generated variable utilized by a CAD file and our program to fast and uniquely access a component. To be emphasized, handle is not an engineering attribute of process plant and merely unique in a single file scope.
- **Type** is the unique identifier for designers to distinguish different components in engineering databases. It is one of the engineering attributes in process plant.
- **Coordinate** is the coordinate of the current component in 3D space.
- **Connections** records the handles of components which has connection relationships with the current component.
- **InsertionCoordinates** records the reference coordinates of connection relationships. As shown in Fig. 3, the dual points (the black dots in Fig. 3) will be coincident on the joint if two components are inter-connected. The insertion coordinate here is the coordinate of either dual point in 3D space. This variable "InsertionCoordinates" is the list of all insertion coordinates related to the current component.

Therefore, we can make use of a CAD document to construct a model's attribute graph.

The procedure of attribute graph construction is conducted as follows. First, we extract all components' handles from the CAD document. Then, each component's attribute values are achieved according to its handle. At last, we add each handle h_i into a graph as a vertex and its engineering attribute as the vertex's attribute. For each handle h_j in *Connections*, we add an edge between h_i and h_j. At this point, the whole procedure of attribute graph construction is completed.

A process plant model can be transformed into an attribute graph accordingly, expressing as

$$M = \{\mathbf{C}, \mathbf{R}\}, \tag{1}$$

where $\mathbf{C} = \{c_1, c_2, \ldots, c_n\}$ is the component set and $\mathbf{R} = \{r_1, r_2, \ldots, r_m\}$ is the connection relationship set. More specifically, a connection relationship $r(c_i, c_j) \in \mathbf{R}$ can be represented as

$$r(c_i, c_j) = (c_i, c_j, A(c_i, c_j)), \tag{2}$$

where $A(c_i, c_j)$ is the engineering attribute of relationship $r(c_i, c_j)$.

2D & 3D models are represented by the same attribute collection through extracting their common attributes. Geometric information is totally discarded during this procedure, thus effectively bypassing the variances in geometry and proportion. For now, process plant 2D & 3D matching degree can be measured by their attribute graphs.

4.3. Relationship type extraction

We propose to exploit type attribute as a component's primary identification. During design procedure, type attribute plays a critical role for designers to differentiate components. Thus, assuming $T(x)$ is the function to get the type of component x, a relationship type $rt(c_i, c_j)$ can be expressed as

$$rt(c_i, c_j) = (T(c_i), T(c_j), A(c_i, c_j)), \tag{3}$$

where $T(c_i)$ and $T(c_j)$ are the types of component c_i and c_j, $A(c_i, c_j)$ is the engineering attribute value of relationship $r(c_i, c_j)$.

Through traversing each model's relationships, all relationship types are extracted as a model's feature and recorded into set $\mathbf{S}$. Thus, a complex CAD model is represented as a set of topology features and the problem of model similarity is boiled down to feature comparison.

4.4. Matching calculation

A 2D engineering drawing could be a global or partial description of a process plant. In light of this, the proposed 2D & 3D matching faces several special situations, listed as follows:

(1) If the connection relationships of a 2D engineering drawing and a 3D model are completely similar, which in other words means they share an identical topology structure, consider them a match;
(2) If partially similar, which indicates the 2D drawing may be a partial description of the 3D model, consider them a match as well;
(3) If satisfies neither (1) or (2), consider them unmatched.

In our work, these situations are used as criteria for judging whether a 2D drawing and a 3D model is matched.

Algorithm 1 gives the details of matching calculation. Assuming $\mathbf{S}_{3D}$ and $\mathbf{S}_{2D}$ are the relationship type sets of 3D model M_{3D} and 2D engineering drawing M_{2D} respectively. In order to determine both partial and global matching likelihood at one shot, we use the model with less relationship types as reference and the other model as target. Let *length* be the smaller size between $\mathbf{S}_{3D}$ and $\mathbf{S}_{2D}$, $\mathbf{V}_{ref}$ and $\mathbf{V}_{tar}$ are the feature vectors of reference and target model, both with a length of *length* and initialized to be 0 filled. In order to generate comparable feature vectors, we traverse every relationship type rt in the reference model and set each component value $\mathbf{V}_{ref}[i]$ $(i = 1, 2, \ldots, length)$ to 1. Meanwhile, if the target model also contains rt, its vector entry $\mathbf{V}_{tar}[i]$ is set to 1 as well. Once the traversal is completed, two compared models are represented by feature vectors $\mathbf{V}_{ref} = (x_1, x_2, \ldots, x_{length})$ and $\mathbf{V}_{tar} = (y_1, y_2, \ldots, y_{length})$. Each index in the feature vector represents a certain relationship type, with the value indicating whether it exists in the corresponding model. At this point, $\mathbf{V}_{ref}$ and $\mathbf{V}_{tar}$ could be input into any similarity metric at hand.

Here we use cosine to cope with binary numbers [38,39], resulting in the following equation:

$$\cos(\mathbf{V}_{ref}, \mathbf{V}_{tar}) = \frac{\sum_{i=1}^{length} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{length} x_i^2 \cdot \sum_{i=1}^{length} y_i^2}}. \tag{4}$$

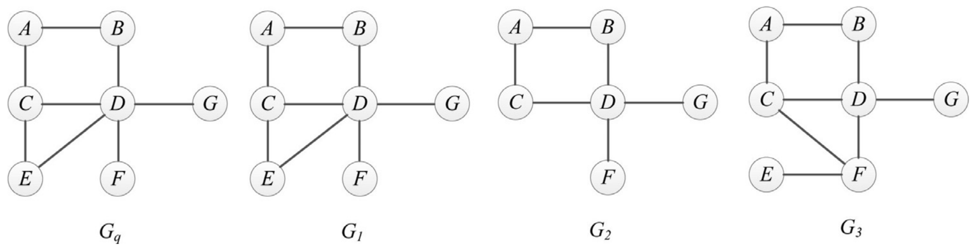


Fig. 8. Model instances. Edges describe relationships and vertices (identified by type attribute) describe components.

Table 1
Feature vectors of models in Fig. 8.

	rt(*A*, *B*)	*rt*(*A*, *C*)	*rt*(*C*, *D*)	*rt*(*B*, *D*)	*rt*(*C*, *E*)	*rt*(*D*, *E*)	*rt*(*D*, *F*)	*rt*(*D*, *G*)	*rt*(*C*, *F*)	*rt*(*E*, *F*)
$\mathbf{S}_{G_q}$	✓	✓	✓	✓	✓	✓	✓	✓		
$\mathbf{S}_{G_1}$	✓	✓	✓	✓	✓	✓	✓	✓		
$\mathbf{S}_{G_2}$	✓	✓	✓	✓			✓	✓		
$\mathbf{S}_{G_3}$	✓	✓	✓	✓			✓	✓	✓	✓
$\mathbf{V}_{G_q}$	1	1	1	1	1	1	1	1		
$\mathbf{V}_{G_1}$	1	1	1	1	1	1	1	1		
$\mathbf{V}_{G_q}$	1	1	1	1			1	1		
$\mathbf{V}_{G_2}$	1	1	1	1			1	1		
$\mathbf{V}_{G_q}$	1	1	1	1	1	1	1	1		
$\mathbf{V}_{G_3}$	1	1	1	1	0	0	1	1		

Algorithm 1 Matching calculation.

Require:
  M_{3D}'s relationship type set $\mathbf{S}_{3D}$;
  M_{2D}'s relationship type set $\mathbf{S}_{2D}$;
Ensure:
  Matching degree $sim(M_{3D}, M_{2D})$;
if $|\mathbf{S}_{2D}| \le |\mathbf{S}_{3D}|$ **then**
  $length = |\mathbf{S}_{2D}|$;
  $\mathbf{S}_{ref} = \mathbf{S}_{2D}$;
  $\mathbf{S}_{tar} = \mathbf{S}_{3D}$;
else
  $length = |\mathbf{S}_{3D}|$;
  $\mathbf{S}_{ref} = \mathbf{S}_{3D}$;
  $\mathbf{S}_{tar} = \mathbf{S}_{2D}$;
end if
$\mathbf{V}_{ref} = [\ 0$ **for** i **from** 1 **to** $length$];
$\mathbf{V}_{tar} = [\ 0$ **for** i **from** 1 **to** $length$];
$i = 1$;
for all $rt \in \mathbf{S}_{ref}$ **do**
  $\mathbf{V}_{ref}[i] = 1$;
  if $rt \in \mathbf{S}_{tar}$ **then**
    $\mathbf{V}_{tar}[i] = 1$;
  end if
  $i{+}{+}$;
end for
$sim(M_{3D}, M_{2D}) = \cos(\mathbf{V}_{ref}, \mathbf{V}_{tar})$;

Finally, the feature similarity between reference model and target model (as well as the matching degree between 2D drawing and 3D model) can be calculated.

Fig. 8 and Table 1 give the visualization and statistics of a calculation example. According to Eq. (1), graphs in Fig. 8 are the attribute graph representations of process plant models, whose feature vectors are presented in Table 1. Based on Algorithm 1, we have the following results:

- $|\mathbf{S}_{G_q}| = |\mathbf{S}_{G_1}| = 8$, so either of M_{G_q} or M_{G_1} can be the reference model. Their matching degree is $sim(M_{G_q}, M_{G_1}) = \cos(\mathbf{V}_{G_q}, \mathbf{V}_{G_1}) = 1.0$;
- $|\mathbf{S}_{G_q}| = 8 > |\mathbf{S}_{G_2}| = 6$, so M_{G_2} should be the reference model. According to M_{G_2}'s relationship types, $sim(M_{G_q}, M_{G_2}) = \cos(\mathbf{V}_{G_q}, \mathbf{V}_{G_2}) = 1.0$;
- $|\mathbf{S}_{G_q}| = |\mathbf{S}_{G_3}| = 8$, so either of M_{G_q} or M_{G_3} can be the reference model. Their matching degree is $sim(M_{G_q}, M_{G_3}) = \cos(\mathbf{V}_{G_q}, \mathbf{V}_{G_3}) = 0.75$.

5. Experiments and discussions

This section will give a validity verification and discussion about our algorithm. Details about our experimental platform are an Intel dual core 2.1 GHz CPU and 3G memory laptop. Our model database includes 2340 pairs of matched 2D & 3D models and 2340 pairs of unmatched 2D & 3D models. Figs. 9–15 show some 3D models and 2D engineering drawings of the database. A typical 3D plant model can generate hundreds of thousands of 2D engineering drawings. For visualization purpose, we randomly select and display a few of them here. All experiments are strictly conducted as described in Section 4. Attribute graph statistics are described in Section 5.1, followed by showcases of the matching results in Section 5.2.

5.1. Attribute graph statistics

Taking CAD documents as input, we use PDSOFT® 3Dpiping and Visual Studio 2008 to automatically extract a model's information (e.g., components, connection relationships, etc.) and transform it into an attribute graph. In Table 2, some graph statistics of the extracted 3D models and 2D drawings are presented, including model types and sizes of their components and relationships.

5.2. Matching results

After constructing attribute graphs, we extract relationship types by using flow direction and insertion coordinate as the attribute respectively. Based on the extracted relationship types, Tables 3 and 4 list some of matching results.

Digging into results in Tables 3 and 4, we can conclude that:

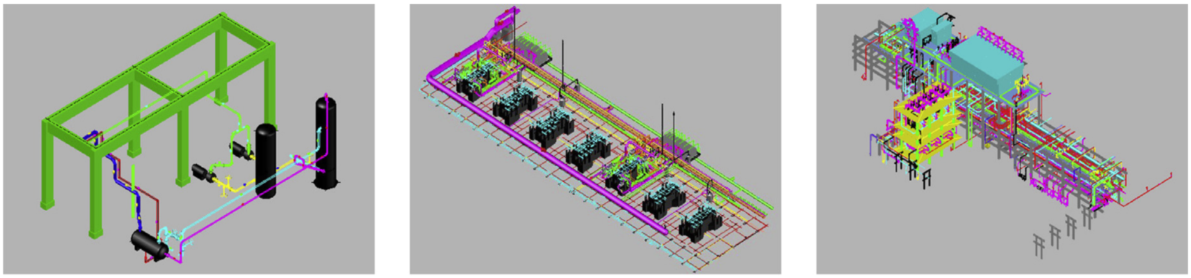
(1) The proposed algorithm can still evaluate whether 2D engineering drawings and 3D models share a common source

Table 2
Attribute graph statistics.

Model	Type	Components $\|\mathbf{C}\|$	Relationships $\|\mathbf{R}\|$
M1 (Fig. 9(a))	3D model	395	393
M2 (Fig. 9(b))	3D model	9795	9728
M3 (Fig. 9(c))	3D model	22,596	22,740
M4 (Fig. 1)	3D model	41,569	41,968
D1 (Fig. 2(a))	Plane drawing of M1	395	393
D2 (Fig. 2(b))	ISO drawing of M1	35	33
D3 (Fig. 10(a))	ISO drawing of M1	67	61
D4 (Fig. 10(b))	ISO drawing of M1	71	70
D5 (Fig. 10(c))	ISO drawing of M1	80	79
D6 (Fig. 10(d))	ISO drawing of M1	111	107
D7 (Fig. 11)	Plane drawing of M2	9795	9728
D8 (Fig. 12(a))	ISO drawing of M2	34	25
D9 (Fig. 12(b))	ISO drawing of M2	55	53
D10 (Fig. 12(c))	ISO drawing of M2	68	33
D11 (Fig. 12(d))	ISO drawing of M2	71	64
D12 (Fig. 13)	Section drawing of M3	22,596	22,740
D13 (Fig. 14(a))	ISO drawing of M3	20	21
D14 (Fig. 14(b))	ISO drawing of M3	22	23
D15 (Fig. 14(c))	ISO drawing of M3	35	35
D16 (Fig. 14(d))	ISO drawing of M3	84	87
D17 (Fig. 15(a))	ISO drawing of M4	37	36
D18 (Fig. 15(b))	ISO drawing of M4	42	41
D19 (Fig. 15(c))	ISO drawing of M4	55	55
D20 (Fig. 15(d))	ISO drawing of M4	103	102

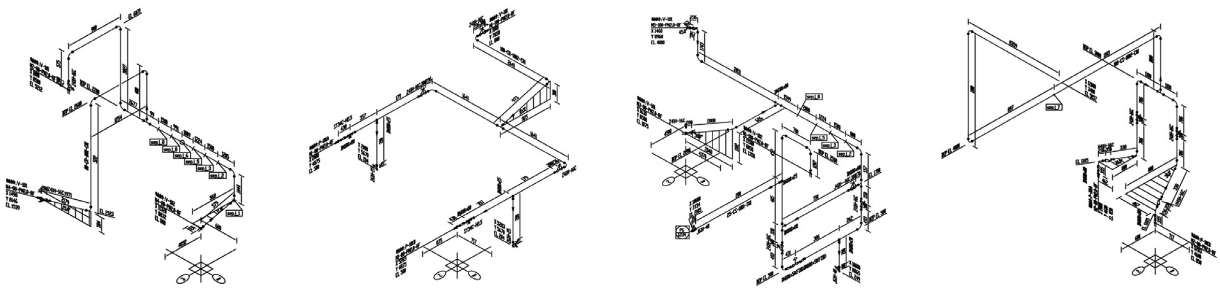
Table 3
Matching results with flow direction.

3D model M_{3D}	2D drawing M_{2D}	Ground truth	$sim(M_{2D}, M_{3D})$	Extraction time	Matching time
M1	D1	Complete Match	1.0	243.6 ms	9.4 ms
	D3	Partial Match	1.0	140.4 ms	0 ms
	D4	Partial Match	1.0	136.8 ms	0 ms
	D5	Partial Match	1.0	168.4 ms	0 ms
	D6	Partial Match	1.0	143.6 ms	6.2 ms
	D7	Unmatch	0.382971	2,748.6 ms	18.4 ms
	D8	Unmatch	0.654654	118.4 ms	6.4 ms
	D12	Unmatch	0.447214	8,109.0 ms	15.6 ms
	D13	Unmatch	0.458831	118.8 ms	6.2 ms
	D17	Unmatch	0	128.2 ms	6.2 ms
M2	D7	Complete Match	1.0	5,235.2 ms	6.2 ms
	D8	Partial Match	1.0	2,624.0 ms	15.8 ms
	D9	Partial Match	1.0	2,717.4 ms	15.8 ms
	D10	Partial Match	1.0	2,653.2 ms	15.6 ms
	D11	Partial Match	1.0	2,644.2 ms	15.8 ms
	D1	Unmatch	0.382971	2,739.2 ms	12.6 ms
	D3	Unmatch	0.527046	2,617.6 ms	12.2 ms
	D12	Unmatch	0	15,309.8 ms	1,133.0 ms
	D13	Unmatch	0.39736	2,617.8 ms	12.4 ms
	D17	Unmatch	0.227314	9,740.4 ms	15.6 ms
M3	D12	Complete Match	1.0	12,723.2 ms	15.8 ms
	D13	Partial Match	0.945905	7,821.8 ms	15.8 ms
	D14	Partial Match	0.919866	7,871.6 ms	15.6 ms
	D15	Partial Match	0.981307	7,743.8 ms	15.6 ms
	D16	Partial Match	0.971008	7,715.4 ms	15.8 ms
	D1	Unmatch	0.447214	8,077.6 ms	15.8 ms
	D3	Unmatch	0.707107	7,747 ms	15.6 ms
	D7	Unmatch	0.227314	9,590.8 ms	25.2 ms
	D8	Unmatch	0.534522	7,903.0 ms	12.4 ms
	D17	Unmatch	0	7,709.6 ms	15.6 ms
M4	D17	Partial Match	1.0	11,793.8 ms	62.2 ms
	D18	Partial Match	0.9759	11,681.4 ms	65.6 ms
	D19	Partial Match	0.974679	11,609.4 ms	62.8 ms
	D20	Partial Match	0.975305	11,637.4 ms	62.8 ms
	D1	Unmatch	0.34641	11,663.0 ms	62.4 ms
	D3	Unmatch	0.333333	11,469.2 ms	65.6 ms
	D7	Unmatch	0.123278	15,943.4 ms	78.0 ms
	D8	Unmatch	0.377964	11,740.8 ms	65.6 ms
	D12	Unmatch	0.115912	18,698.0 ms	78.0 ms
	D13	Unmatch	0	11,491.4 ms	68.8 ms



(a) A simple process plant model. (b) An urea production line plant model. (c) A petrol etherification plant model.

Fig. 9. Process plant 3D models.



(a) (b) (c) (d)

Fig. 10. ISO drawings of 3D model in Fig. 9(a). These drawings describe the different pipelines which come from the same plant.

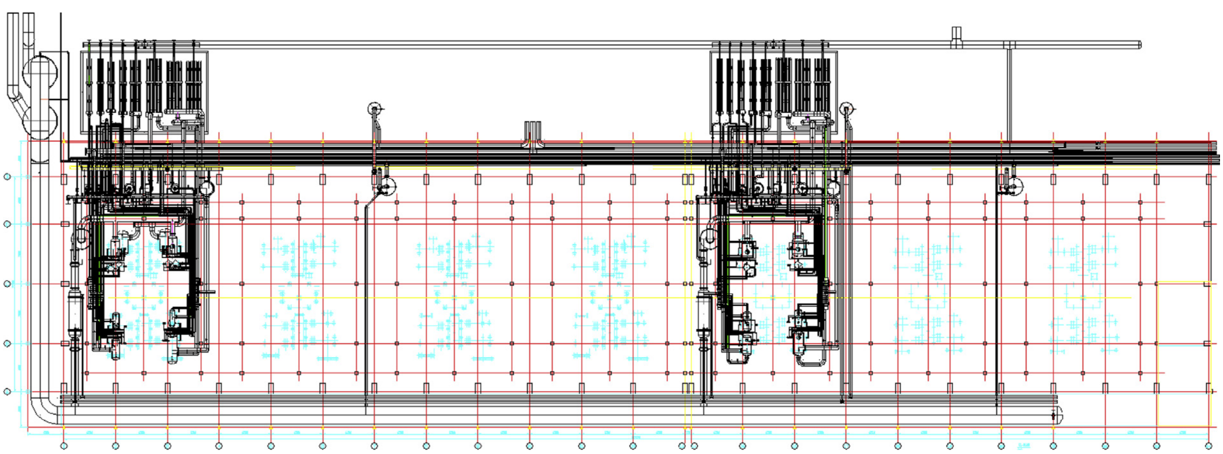


Fig. 11. Plane drawing without annotations and illustrations of 3D model in Fig. 9(b).

plant, even they have differences in content structure. In Tables 3 and 4, high similarities are observed in both partial matched pairs and complete matched pairs.

(2) Differences in graphical representation do not affect our experimental results. Even though the graphical representations of 2D drawings and 3D models are different, the matching results in Tables 3 and 4 are realistic;

(3) Our algorithm can recognize unmatched models. As indicated in Table 2, D1 is not a 2D drawing of 3D model M2. Accordingly, their matching result in Table 4 is $sim(M2, D1) = 0$.

(4) Most calculation time of our experiment lies in a proper range. Both the extraction and matching calculation in Table 3 and Table 4 are finished within an acceptable time.

5.3. Performance analysis

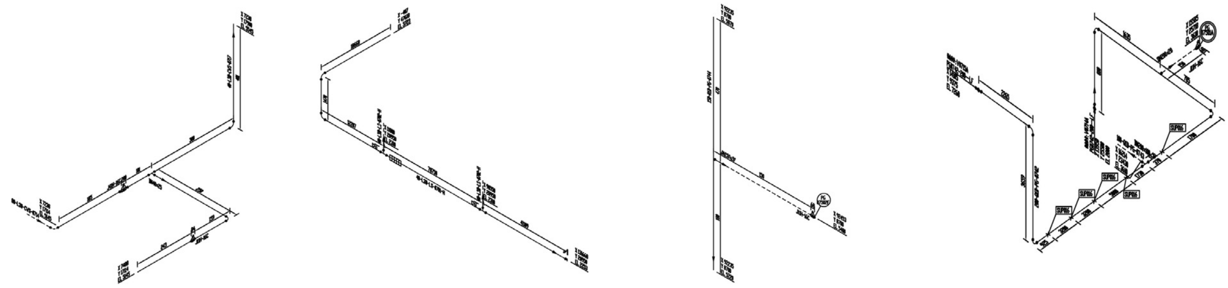
We first introduce a group of terms and metrics before giving any quantitative analysis.

- True Positives (TP) are positive samples that are correctly labeled as positives.
- False Positives (FP) are negative samples that are incorrectly labeled as positive.
- False Negatives (FN) are positive samples that are incorrectly labeled as negative.
- True Negatives (TN) are negative samples that are correctly labeled as negative.

Recall is the proportion of positive samples that are correctly labeled. It measures an algorithm's ability in recognizing the rare positive samples (because they are the ones we care most about). It is defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{5}$$

Accuracy is the proportion of all samples that are correctly labeled. It measures an algorithm's ability in classification among all



(a) (b) (c) (d)

Fig. 12. ISO drawings of 3D model in Fig. 9(b).

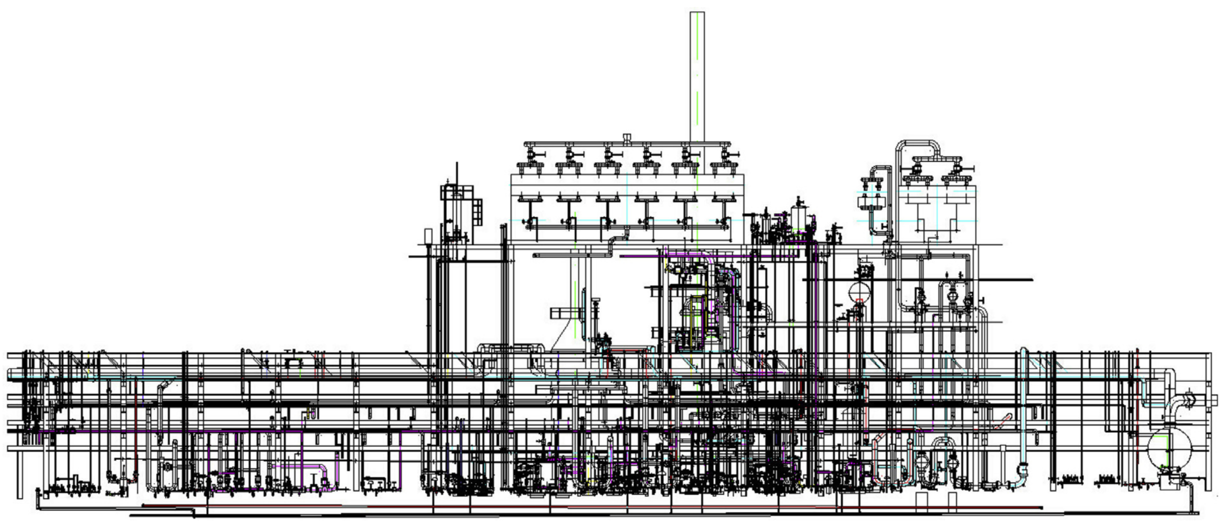
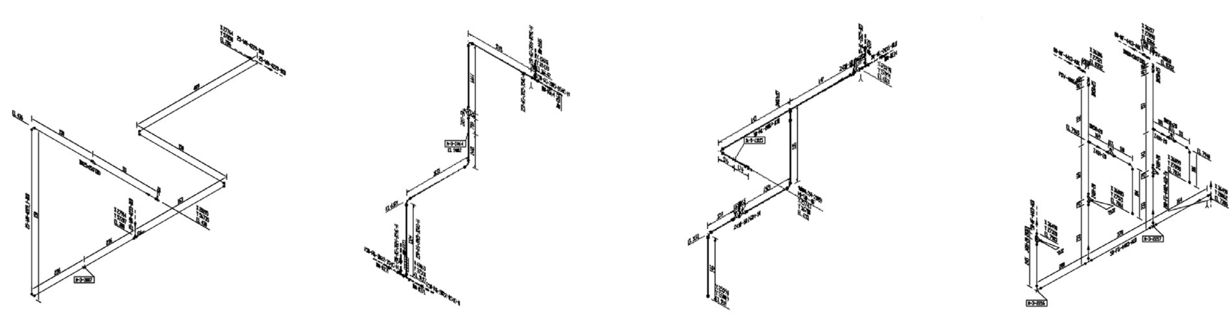
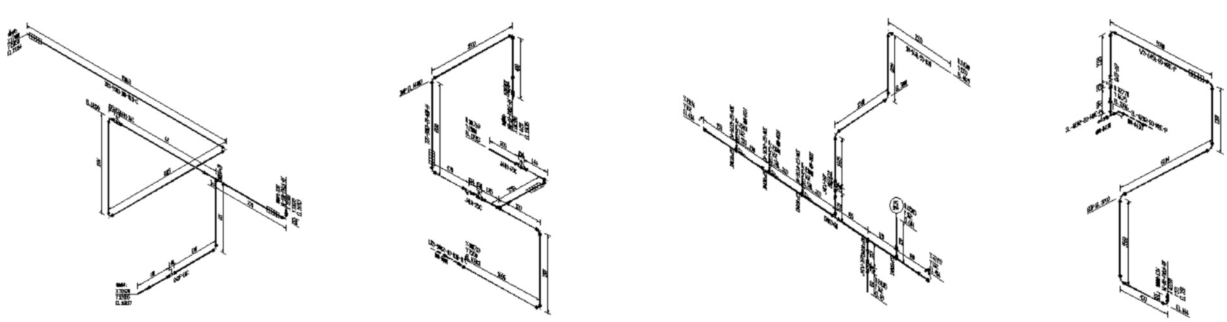


Fig. 13. Plane drawing without annotations and illustrations of 3D model in Fig. 9(c).



(a) (b) (c) (d)

Fig. 14. ISO drawings of 3D model in Fig. 9(c).



(a) (b) (c) (d)

Fig. 15. ISO drawings of 3D model in Fig. 1.

Table 4
Matching results with insertion coordinate.

3D model M_{3D}	2D drawing M_{2D}	Ground truth	$sim(M_{2D}, M_{3D})$	Extraction time	Matching time
M1	D1	Complete Match	1.0	306.0 ms	15.8 ms
	D2	Partial Match	1.0	171.6 ms	15.6 ms
	D3	Partial Match	1.0	175.0 ms	12.4 ms
	D4	Partial Match	1.0	178.0 ms	9.6 ms
	D5	Partial Match	1.0	199.6 ms	15.8 ms
	D6	Partial Match	1.0	193.4 ms	15.8 ms
	D7	Unmatch	0	4,377.4 ms	330.6 ms
	D8	Unmatch	0	175.0 ms	15.4 ms
	D12	Unmatch	0	8,966.6 ms	270.4 ms
	D13	Unmatch	0	149.2 ms	15.2 ms
	D17	Unmatch	0	183.8 ms	15.8 ms
M2	D7	Complete Match	1.0	7,605.2 ms	327.6 ms
	D8	Partial Match	1.0	4,106.2 ms	302.4 ms
	D9	Partial Match	1.0	4,134.2 ms	302.8 ms
	D10	Partial Match	1.0	4,162.0 ms	302.8 ms
	D11	Partial Match	1.0	4,140.4 ms	299.6 ms
	D1	Unmatch	0	4,395.8 ms	334.0 ms
	D2	Unmatch	0	4,121.4 ms	308.8 ms
	D12	Unmatch	0	15,309.8 ms	1,133.0 ms
	D13	Unmatch	0	4,124.4 ms	296.4 ms
	D17	Unmatch	0	4,102.8 ms	305.8 ms
M3	D12	Complete Match	1.0	18,692.2 ms	830.0 ms
	D13	Partial Match	1.0	10,137.2 ms	767.6 ms
	D14	Partial Match	0.955533	10,121.6 ms	773.8 ms
	D15	Partial Match	0.985611	10,162.2 ms	764.2 ms
	D16	Partial Match	1.0	10,171.0 ms	764.8 ms
	D1	Unmatch	0	10,527.0 ms	801.8 ms
	D2	Unmatch	0	10,186.8 ms	780.0 ms
	D7	Unmatch	0	14,886.0 ms	1,141.8 ms
	D8	Unmatch	0	10,149.2 ms	770.6 ms
	D17	Unmatch	0	10,127.6 ms	767.4 ms
M4	D17	Partial Match	1.0	19,575.2 ms	1,488.2 ms
	D18	Partial Match	1.0	19,992.8 ms	1,547.4 ms
	D19	Partial Match	1.0	19,771.4 ms	1,538.0 ms
	D20	Partial Match	1.0	19,596.8 ms	1,519.4 ms
	D1	Unmatch	0	19,770.6 ms	1,556.8 ms
	D2	Unmatch	0	19,522.0 ms	1,510.0 ms
	D7	Unmatch	0	24,532.8 ms	1,890.6 ms
	D8	Unmatch	0	19,615.6 ms	1,532.0 ms
	D12	Unmatch	0	30,963.2 ms	2,421.2 ms
	D13	Unmatch	0	20,002.4 ms	1,497.2 ms

classes, i.e., to tell positive from negative. It is defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{6}$$

Precision is the proportion of samples classified as positive that are also truly positive. Different from accuracy, it only measures an algorithm's classification ability for the positive class. It is defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{7}$$

Now given a threshold φ, we consider M_{2D} matches with M_{3D} if $sim(M_{2D}, M_{3D}) \geq \varphi$. The φ-recall/accuracy curves with different attributes are shown in Fig. 16. We can have a clear visualization on how the metrics fluctuate with different φ. In our experiment, both accuracy and recall peak near $\varphi = 0.8$ in all these figures. In real-world applications, we can use a validation dataset to determine the value of φ that yields the best performance.

For comparison, this proposed algorithm has a clear performance advantage over the Min-Edit Distance based algorithm [16] according to the recall-precision curves in Fig. 17. An ideal recall-precision curve leads to be in the upper-right-hand corner [40]. Apparently, both recall-precision curves of this work (the blue lines) are consistently in conformity with that trend, no matter what attribute is being used. The curve of Min-Edit Distance based algorithm is also in the upper-right-hand corner when using insertion coordinate as attribute. However, it tends to be in the lower-left-hand corner when attribute is flow direction. In this case, the choice of attribute can greatly affect the final performance. Sometimes we may even have to spare huge human efforts to decide which attributes are feasible, resulting in undoubted low efficiency.

Table 5
The confusion matrix of our matching results. The relationship type's attribute is insertion coordinate and $\varphi = 0.8$.

		Predicted	
		Negative	Positive
Actual	Negative	2340 (TN)	0 (FP)
	Positive	10 (FN)	2330 (TP)

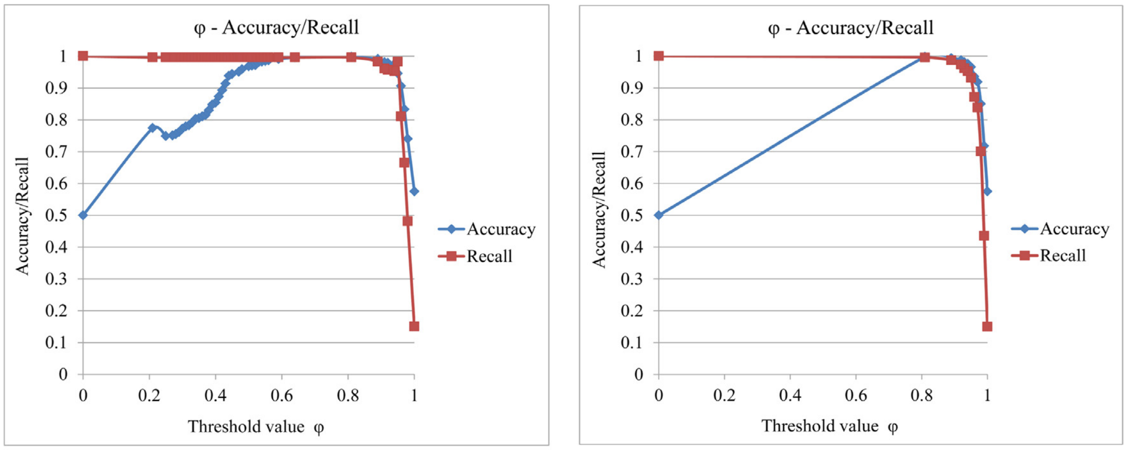
For quantitative analysis, Table 5 lists the confusion matrix of matching results with insertion coordinate when $\varphi = 0.8$. Recall and accuracy are 99.6% and 99.8%, respectively according to this table. These statistics suggest that our experimental results are mostly in accordance with the ground truth.

In conclusion, performance of the proposed work in this study has a remarkable improvement in terms of flexibility and stability.

5.4. Discussions

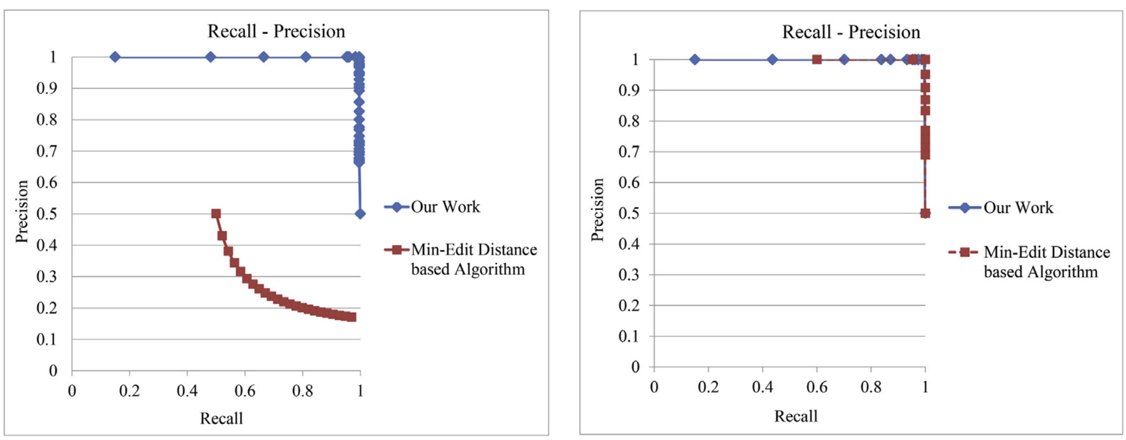
5.4.1. Attribute selection

The choice of attribute in Eq. (3) influences the matching degrees. Table 6 shows the quantities of relationship types for each model by using insertion coordinate and flow direction as attribute



(a) Flow direction. (b) Insertion coordinate.

Fig. 16. φ-recall/accuracy curves with different attributes.



(a) Flow direction. (b) Insertion coordinate.

Fig. 17. Recall-precision curves of our work and the Min-Edit Distance based algorithm with different attributes.

Table 6
Statistics of relationship types.

Model	Relationships $\lvert\mathbf{R}\rvert$	Relationship types $\lvert\mathbf{S}\rvert$	
		Insertion coordinate	Flow direction
M1	393	393	75
M2	9728	9728	329
M3	22,740	22,740	521
M4	41,968	41968	2548
D1	395	393	75
D2	35	33	22
D3	61	61	18
D4	70	70	24
D5	79	79	19
D6	107	107	28
D7	9795	9728	329
D8	34	25	14
D9	55	53	39
D10	68	33	21
D11	71	64	48
D12	22,740	22,740	521
D13	21	21	19
D14	23	23	13
D15	35	35	27
D16	87	87	35
D17	36	36	18
D18	41	41	21
D19	55	55	20
D20	102	102	41

respectively. It can be seen that the number of relationship types is equal to the quantity of relationships when choosing insertion coordinate. In other words, the relationship types can wholly represent a model's connection relationships. According to Table 4, most matching results of matched pairs are 1.0 and unmatched pairs are 0. These results suggest that the proposed algorithm can yield more accurate results with insertion coordinate.

On the other hand, relationships outnumber relationship types when using flow direction as attribute. According to Eq. (3), different relationships may yield a single identical relationship type, because flow direction is enumerable within only three possible values: 0 – no direction, 1 – inflow and 2 – outflow. Consequently, the relationship types cannot accurately represent a model and performance sees some decline comparing to that of insertion coordinate. As indicated in Table 3, similarities are relatively higher between unmatched pairs, even though we can still use a predefined threshold to determine whether they are matching.

5.4.2. Complexity discussion

Let us be given a 3D model $M_{3D} = \{\mathbf{C}_{3D}, \mathbf{R}_{3D}\}$ and a 2D engineering drawing $M_{2D} = \{\mathbf{C}_{2D}, \mathbf{R}_{2D}\}$. Their relationship type sets are $\mathbf{S}_{3D}$ and $\mathbf{S}_{2D}$ respectively. For each model, the complexity of relationship type extraction is linear to the number of relationships, i.e., $O(|\mathbf{R}_{nD}|)$ $(n \in \{2, 3\})$. The complexity of matching calculation is $O(\min\{|\mathbf{S}_{3D}|, |\mathbf{S}_{2D}|\})$. That is to say, the time costs of extraction and matching calculation would both be growing with the increasing of relationships.

5.4.3. Robustness discussion

The proposed algorithm has a great property of being geometric deformation invariant. Proportion methods and graphics library would make components' 2D geometries different from their 3D shape projections. Nonetheless in our proposed algorithm, all these unnecessary and extra variances are discarded by using topology features. The recall-precision curves in Fig. 17 is a convincing showcase of the outcome.

5.5. Performance comparison

The presented algorithm is more applicable in real engineering applications comparing to the Min-Edit Distance based algorithm (see Section 3.4).

In terms of usability, users can choose any suitable attribute according to their measuring emphasis. In the Min-Edit Distance based algorithm, insertion coordinate is the only feasible attribute allowed. Unlike the one-to-one correspondence in coordinates, other engineering attributes (such as flow direction) have much narrower value ranges than that of relationships. So it would be difficult to find the exact correspondences from the many overlaps. In this case, computing min-edit distance becomes NP-hard and the algorithm run time goes indefinitely long (sometimes no results yielded within an acceptable time range). On the contrary, in this study, other engineering attributes besides coordinate information can also be exploited as a relationships attribute (see Table 3), endowing it with more scalability and flexibility.

In terms of performance, the presented algorithm in this paper has a prominent advantage. We conducted several sets of comparison experiments using different attributes to demonstrate their differences. For the Min-Edit Distance based algorithm, we randomly assume it's matching or not when calculation exceeds a prescribed time during experiment, which in other word is a 50% precision. Here we give two results with attributes being flow direction and coordination respectively. As shown in Fig. 17(a), the Min-Edit Distance based algorithm's recall-precision curve (the red line) is in the lower-left-hand corner when the attribute is flow direction. This suggests the algorithm performance faces enormous challenges. Conversely, as illustrated in Fig. 17(a) and (b), the recall-precision curves of this presented algorithm (the blue lines) remain stable and excellent (the upper-right-hand corner) in both cases, proving its outstanding performance.

6. Conclusion, limitation and future work

In this paper, we propose a novel algorithm to solve the 2D & 3D matching problem in process plant. This algorithm first transforms 2D engineering drawings and 3D models into attribute graphs by making use of CAD documents; secondly, each model's feature vector is generated through extracting its relationship types; lastly, vector similarity is calculated to measure the matching degree between two corresponding CAD models. The proposed algorithm is simple and embraces a positive property of geometric deformation invariant. Experiments demonstrate that our proposed algorithm yields realistic results within a proper time cost and meets engineering application standards.

The procedure of matching calculation is executed pairwise, so that the feature vectors are not uniquely defined and perhaps cannot be used for a fast model retrieval. However, the problem can be solved by several existing engineering techniques such as distributed computation and inverted indexing. In the following work, we are attempting to overcome this problem and then construct an integrative retrieval system, which can retrieve relevant 3D models and 2D engineering drawings at the same time.

Acknowledgments

Our work is supported by the National Natural Science Foundation of China (No. 61300160). The process plant models used in this paper are courtesy of Beijing Zhongke Fulong Computer Technology Co., Ltd.

References

[1] N. Ishii, T. Fuchino, M. Muraki, Life cycle oriented process synthesis at conceptual planning phase, Comput. Chem. Eng. 21 (10) (1997) S953–S958.
[2] S. Rao, The Research and Implementation of Engineering Drawing Automatic Generation, Ph.D. thesis, Beijing: Institute of Computing Technology, Chinese Academy of Sciences, 2006.
[3] J. Bai, S. Gao, W. Tang, Y. Liu, S. Guo, Design reuse oriented partial retrieval of CAD models, Comput.-Aided Des. 42 (12) (2010) 1069–1084.
[4] S. Biasotti, S. Marini, M. Spagnuolo, B. Falcidieno, Sub-part correspondence by structural descriptors of 3D shapes, Comput.-Aided Des. 38 (9) (2006) 1002–1019.
[5] S. Biasotti, A. Cerri, A. Bronstein, M. Bronstein, Recent trends, applications, and perspectives in 3D shape similarity assessment, Comput. Graphics Forum 35 (6) (2015) 87–119.
[6] A. Ferreira, S. Marini, M. Attene, M.J. Fonseca, M. Spagnuolo, J.A. Jorge, B. Falcidieno, Thesaurus-based 3D object retrieval with part-in-whole matching, Int. J. Comput. Vis. 89 (2) (2010) 327–347.
[7] Z. Liu, S. Bu, K. Zhou, S. Gao, J. Han, J. Wu, A survey on partial retrieval of 3D shapes, J. Comput. Sci. Technol. 28 (5) (2013) 836–851.
[8] O. Van Kaick, H. Zhang, G. Hamarneh, D. CohenOr, A survey on shape correspondence, Comput. Graphics Forum 30 (6) (2011) 1681–1707.
[9] M. Wang, Y. Gao, K. Lu, Y. Rui, View-based discriminative probabilistic modeling for 3D object retrieval and recognition, IEEE Trans. Image Process. 22 (4) (2013) 1395–1407.
[10] M. Soo Bae, I.K. Park, Content-based 3D model retrieval using a single depth image from a low-cost 3D camera, Visual Comput. 29 (6–8) (2013) 555–564.
[11] Z. Lahner, E. Rodola, F.R. Schmidt, M.M. Bronstein, D. Cremers, Efficient globally optimal 2D-to-3D deformable shape matching, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016.
[12] W. Mohamed, A.B. Hamza, Reeb graph path dissimilarity for 3D object matching and retrieval, Visual Comput. 28 (3) (2012) 305–318.
[13] L. Li, H. Wang, T.-J. Chin, D. Suter, S. Zhang, Retrieving 3D CAD models using 2D images with optimized weights, in: 2010 3rd International Congress on Image and Signal Processing (CISP), 4, 2010, pp. 1586–1589.
[14] J. Pu, K. Lou, K. Ramani, A 2D sketch-based user interface for 3D CAD model retrieval, Comput.-Aided Des. Appl. 2 (6) (2005) 717–725.
[15] R. Wen, W. Tang, Z. Su, Measuring 3D process plant model similarity based on topological relationship distribution, Comput. Aided Des. Appl. 14 (04) (2017) 422–435.
[16] R. Wen, W. Tang, Z. Su, A 2D engineering drawing and 3D model matching algorithm for process plant, in: International Conference on Virtual Reality and Visualization, 2015, pp. 154–159.
[17] A. Burdorf, B. Kampczyk, M. Lederhose, H. Schmidt-Traub, CAPD-computer-aided plant design, Comput. Chem. Eng. 28 (1–2) (2004) 73–81.
[18] J.W. Tangelder, R.C. Veltkamp, A survey of content based 3D shape retrieval methods, Multimed. Tools Appl. 39 (3) (2008) 441–471.
[19] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, Skeleton based shape matching and retrieval, in: Shape Modeling International, 2003, p. 130.
[20] A. Tal, E. Zuckerberger, Mesh retrieval by components, Commun. Comput. Inf. Sci. 4 (2004) 44–57.
[21] M.F. Demirci, R.H.V. Leuken, R.C. Veltkamp, Indexing through laplacian spectra, Comput. Visual Image Understanding 110 (3) (2008) 312–325.
[22] G.K.L. Tam, R.W.H. Lau, Deformable model retrieval based on topological and geometric signatures, IEEE Trans. Visualization Comput. Graphics 13 (3) (2007) 470–482.
[23] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, H. Zhang, Contextual part analogies in 3D objects, Int. J. Comput. Vis. 89 (2) (2010) 309–326.
[24] Y. Kleiman, O. Van Kaick, O. Sorkine-Hornung, D. Cohen-Or, SHED: shape edit distance for fine-grained shape similarity, ACM Trans. Graph 34 (6) (2015) 1–11.
[25] Z. Su, L. Zhou, W. Li, Y. Dai, W. Tang, Topology authentication for CAPD models based on laplacian coordinates, Comput. Graph 37 (4) (2013) 269–279.
[26] S.M. Jeon, J.H. Lee, G.J. Hahm, H.W. Suh, Automatic CAD model retrieval based on design documents using semantic processing and rule processing, Comput. Ind. 77 (2016) 29–47.
[27] Z. Zeng, A.K. Tung, J. Wang, J. Feng, L. Zhou, Comparing stars: on approximating graph edit distance, Proc. VLDB Endowment 2 (1) (2009) 25–36.
[28] W. Zheng, L. Zou, X. Lian, D. Wang, D. Zhao, Graph similarity search with edit distance constraint in large graph databases, in: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management, ACM, 2013, pp. 1595–1600.

[29] S. Lagraa, H. Seba, R. Khennoufa, H. Kheddouci, et al., A distance measure for large graphs based on prime graphs, Pattern Recognit. 47 (9) (2014) 2993–3005.
[30] J.W. Raymond, E.J. Gardiner, P. Willett, Rascal: calculation of graph similarity using maximum common edge subgraphs, Comput. J. 45 (6) (2002) 631–644.
[31] D. Koutra, Large Graph Mining and Sense-making, Ph.D. thesis, Microsoft Research, 2014.
[32] X. Dai, The Research on AEC CAD Modeling Technology Based on Enhanced Graph and Multi-state Model, Ph.D. thesis, Beijing: Institute of Computing Technology, Chinese Academy of Sciences, 2000.
[33] Z. Su, W. Li, J. Kong, Y. Dai, W. Tang, Watermarking 3D CAPD models for topology verification, Comput.-Aided Des. 45 (7) (2013) 1042–1052.
[34] R. Huang, S.S. Zhang, H.T. Fan, J. Tao, A matching algorithm between precursory 3D process model and 2D working procedure drawing based on subgraph isomorphism, Sci. China Technol. Sci. 54 (7) (2011) 1826–1832.
[35] M. Stephan, J.R. Cordy, A survey of model comparison approaches and applications, in: International Conference on Model-Driven Engineering and Software Development, 2013, pp. 265–277.
[36] H. Bunke, On a relation between graph edit distance and maximum common subgraph, Pattern Recognit. Lett. 18 (9) (1997) 689–694.
[37] A. Fischer, C.Y. Suen, V. Frinken, K. Riesen, H. Bunke, Approximation of graph edit distance based on hausdorff matching, Pattern Recognit. 48 (2) (2015) 331–343.
[38] Y. Zhang, Y.D. Liu, J.I. Zhao, Vector similarity measurement method, Tech. Acoust. 28 (4) (2009) 532–536.
[39] S.H. Cha, Comprehensive survey on distance/similarity measures between probability density functions, Int. J. Math. Models Methods Appliedences 1 (2007).
[40] J. Davis, M. Goadrich, The relationship between Precision-Recall and ROC curves, in: International Conference on Machine Learning, 2010, pp. 233–240.