IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Partial Matching of Large Scale Process Plant Models Using Random Walk on Graphs

WEIWEI MAO[1], ZHUHENG LU[1], YUEWEI DAI[2], WEIQING LI[3], AND ZHIYONG SU[1]
[1]School of Automation, Nanjing University of Science and Technology, NanJing 210094, China
[2]School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, NanJing 210044, China
[3]School of Computer Science and Engineering, Nanjing University of Science and Technology, NanJing 210094, China

Corresponding author: Zhiyong Su (e-mail: su@ njust.edu.cn).

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1004904, the Fundamental Research Funds for the Central Universities under Grant 30918012203, and the Pre-research Project of The 13th Five Year Plan under Grant 61409230104 and 315100104.

ABSTRACT 3D process plant models(PPMs) in the process industry normally consists of thousands of components. And, there are many similar local structures in the PPM. Due to the complex process flow, the topology relationship among components is very complicated. Therefore, designing a new PPM is quite time consuming. In order to shorten the design cycle, content based model retrieval for PPMs is an imperative requirement. In this paper, we propose a partial matching framework for PPMs based on graph matching aiming at improving design efficiency and realizing design reuse. The random walk algorithm is employed to distinguish similar local structures. Specifically, each PPM is represented by an undirected labeled graph. The local topological feature of each component is extracted based on the random walk algorithm. For partial matching, a subgraph isomorphism algorithm is introduced. The matching process is accelerated by using the local topological feature to generate an optimized initial state and alleviate the computation of feasible rules. Experimental results show the feasibility and effectiveness of our matching framework.

INDEX TERMS graph matching, model retrieval, partial matching, topological feature, random walk

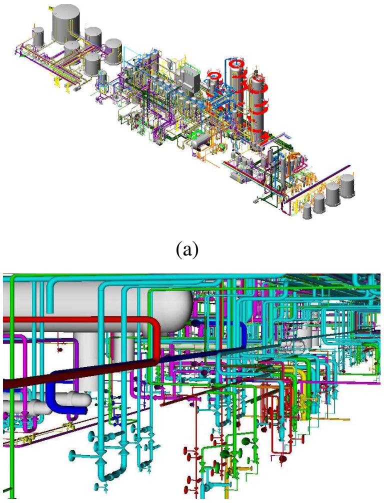
I. INTRODUCTION

Computer aided process plant design plays an important role in the early stage of a process plant's life cycle which consists of planning, design, construction, operation, and disposal [1]. Designing a new process plant model(PPM) is currently quite time consuming due to the complexity of the PPM. In order to shorten the design cycle and improve efficiency, reuse of existing examined PPMs is often required. For this purpose, a matching method between existing PPMs and input partial PPM is necessary as this strategy provides design reference. In addition, reuse of existing examined PPMs can avoid unnecessary mistakes and guarantee the quality of PPMs.

The PPM contains complex diverse information including geometry information, topology information, and engineering information. Geometry information describes the shape of each component. Topological information refers to the adjacency relationships among all components. Engineering information characterizes the design constraints, engineering disciplines, etc.

To some extent, a PPM can be viewed as an assembly model with a much larger number of components. As shown in Fig. 1, a process plant model usually consists of thousands of components in various shapes.

- Components. A PPM is comprised of basic components, including equipment, valves, pipes, etc. Different from general 3D models and computer aided design(CAD) models represented by meshes or surfaces, components of the PPM are designed by basic entities such as cylinder, sphere, box, wedge, etc. Consequently, components are represented by type attributes and geometric parameters, rather than meshes or surfaces in 3D&CAD models.
- Topological information. The topology structure is the kernel of the PPM. The PPM's topology concentrates on the most economical spatial arrangement and interconnections among components. The connections should satisfy the engineering rules such as geometry param-


(a)

(b)

FIGURE 1. Two examples of the 3D PPM.

eters, flow directions, and so on. Connection points are one kind of topology modeling method for PPMs. Each connection point contains information describing the geometry parameters and engineering constraints.

The PPM normally contains lots of similar local structures besides a large number of components, such as the combination of pipe, gasket, and different valves. Considering the PPM, even a small difference between two local structures probably implies two different process flows. Distinguishing similar local structures is propitious to differentiate process flows. Therefore, the matching of PPMs should focus on features that reflect topological relationships among components rather than the shape features of components. Features of PPMs should have the ability to identify similar local structures.

To the best of our knowledge, few published works deal with the partial matching of PPMs. Wen *et al.* [2] used the topological relationship distribution to measure the overall similarity between different types of PPMs. How to discover the target partial model quickly in the complex PPM under the interference of similar local structure has not been studied. Current content-based 3D&CAD model matching methods mainly focus on shape similarity between models [3], [4]. The extracted features are compressed shape representation of 3D&CAD models, and the results mainly reveal the shape similarities. Some other works attempt to solve the matching problem between assembly models [5], [6]. Both shape features and assembly information are used to perform global matching between assembly models. However, as for PPMs, topology matching is much more concerned compared with shape similarity. Topology relationship among components indicates the structure information, and even the process flow. Furthermore, existing matching algorithms of assembly models only deal with a small number of components, which suffers high computational burden when facing a large number of components. In conclusion, it is necessary to develop partial matching methods for PPMs.

In this paper, we proposed a partial matching framework for large scale PPMs based on random walk(RW) on graphs. The PPM is defined as an undirected labeled graph preserving its topological relationship. In order to distinguish similar local structures and thereby accelerate the matching process, the local topological feature of each component is computed based on the RW on the extracted subgraph. As exact partial matching between PPMs should focus on the topological relationship matching, a subgraph isomorphism algorithm is introduced to perform partial matching. The local topological feature is used to accelerate the matching process. The main contributions of this paper are summarized as follows:

- A method for distinguishing similar local structures based on the RW algorithm is proposed. The local topology structure centered on each component is extracted at first, then the local topological feature of each component used for distinguishing similar local structures is extracted based on the RW on the local topology structure.
- A partial matching algorithm of PPMs based on VF2 algorithm combined with local topological features is proposed. VF2 subgraph isomorphism algorithm is introduced as the PPM is represented by an undirected labeled graph. In order to accelerate the matching process, the local topological features are used for generating an optimized initial state and alleviating the computing overhead.

The rest of this paper is organized as follows. In section II, we give a brief introduction to related works including 3D&CAD model matching and assembly model matching. In section III, we propose our partial matching framework for PPMs. Section IV shows the experimental results and validates the effectiveness of this method.

II. RELATED WORK

In this section, we give a brief review of the state-of-the-art general 3D&CAD models matching methods and assembly model matching methods. To the best of our knowledge, only a few published works attempt to solve the PPM matching problem. Since the RW algorithm is introduced for local topological feature extraction, some applications of the RW algorithm are reviewed at first.

A. RANDOM WALK ALGORITHM

Random Walk is a stochastic diffusion process. It has a close relationship with many phenomena in daily life, such as Brownian Motion, odor diffusion, etc. Statistics of the random walk process, such as average first pass time and steady-state probability, have been widely used in information retrieval [7], graph matching [8], image segmentation [9]–[11], video supervoxels [12], etc. One famous application is PageRank [13].

B. 3D MODEL MATCHING

General 3D models are polygon representations of objects, such as chairs, tables, creatures, etc. According to the types of features employed, existing general 3D model matching techniques can be divided into four categories:

- Geometry based methods. Geometry based techniques utilize the distribution of geometric elements of a 3D model to perform matching. Osada *et al.* [14] designed shape functions over random surface points to get a continuous probability as the representation for 3D models. Ankerst *et al.* [15] introduced 3D shape histograms as an intuitive and powerful similarity model for 3D objects. Sipiran *et al.* [16] presented a data-aware partition method from meshes and performed matching by solving Integer Quadratic Program problem. Zou *et al.* [17] proposed a combined shape distribution(CSD) descriptor for 3D model retrieval based on principle plane analysis and group integration. Zhou and Zeng [18] extracted three new features based on compressive sensing and use supervised learning to determine the weighting coefficients for these features. Luciano and Hamza [19] used stacked sparse autoencoders to learn deep shape descriptors from geodesic moments of 3D shapes.
- Graph based methods. Graph based methods attempt to perform matching by using topological information between shape components. In this case, 3D models are often represented by Reeb graphs or skeletons. Hilaga *et al.* [20] first construct multiresolution Reeb graphs to represent 3D models. Barra and Biasotti [21] proposed extended Reeb graphs(ERGs) to describe 3D models and use kernel functions on ERGs to measure the similarity. Sirin and Demirci [22] proposed an enhanced skeleton representation of the 3D model and used Earth Movers' Distance algorithm to perform skeleton matching.
- View based methods. View based methods use a set of 2D images captured from different views to represent a 3D model. Recent 3D model retrieval methods mainly focus on view-based techniques [23], [24]. Hao *et al.* [25]extracted visual features for multi-view representation and constructed the graphical model to represent the individual 3D model. Guo *et al.* [26] used a deep embedding network to extract deep features from 2D images of 3D models and formulate retrieval task as a set-to-set matching problem. Sfikas *et al.* [27] proposed a convolutional neural network based method for the creation of 3D model descriptors. The proposed network exploits the 2D panoramic view representation of 3D models as input. Liu *et al.* [28] proposed a single image-based 3D model retrieval method. With a few semantic bounding boxes in the image, the method retrieved the most similar 3D models.
- Spectral geometry based methods. The surface of a 3D model can be interpreted as a manifold. Spectral geometry based methods extract features based on the Laplace-Beltrami operator on the manifold, such as heat kernel signature [29] and wave kernel signature [30]. Recently, some matching algorithms based on spectral geometry are proposed for non-rigid and deformable 3D model retrieval [31], [32].

C. CAD MODEL MATCHING

Reuse of mechanical CAD models is of great significance in the manufacturing industry. Different from general 3D models, CAD models which often represented by boundary representation(B-rep) or constructive solid geometry(CSG), are comprised of surfaces and curves instead of meshes. There are some efforts attempting to solve the matching problem of CAD models by using the general 3D model matching algorithms. Hong *et al.* [33] employed shape distribution graph for overall CAD shape comparison. Kim [4] also adopted shape distribution features for mechanical CAD model retrieval.

More generally, most matching approaches for CAD models focus on extracting features from B-rep or CSG description of CAD models. B-rep is a method for representing shapes using the limits. Face adjacency graph(FAG) is a popular description method of the CAD model which can be easily generated from the B-rep model. Huangfu *et al.* [34] proposed Labeled Attribute Adjacency Graphs (LAAGs) to represent CAD models and a spatial bag of words for CAD model matching. Tao *et al.* [35] proposed a retrieval method for CAD solid models based on the segmentation of FAGs.

Semantic matching of CAD models is also of great importance as CAD models possess rich manufacturing semantics. Cardone *et al.* [36] proposed a CAD model retrieval method based on machining features. Qin *et al.* [37] proposed a fine granularity semantic descriptor for CAD model and sketch matching.

D. ASSEMBLY MODEL MATCHING

The assembly model is a model that consists of individual part models [38]. The assembly relationship between partial models is considered more than the shape feature. Deshmukh *et al.* [38] addressed this problem by utilizing assembly statistics and mutual relationships. Chen *et al.* [5] proposed a new assembly retrieval approach for flexible input queries. Kim *et al.* [6] extracted shape features of partial models and the assembly relationships between partial models to realize the assembly model retrieval. Lupinetti *et al.* [39] proposed an assembly descriptor for an assembly CAD model, called Enriched Assembly Model(EAM). Matching between two assembly models is transformed into the sub-isomorphism problem between two EAMs. As for PPMs, Wen *et al.* [2] used topological relationship distribution to measure the similarity between different types of PPMs. Rantala *et al.* [40] proposed graph simplification techniques to get a simple model from the 3D PPM. Several graph matching algorithms were adopted to perform matching.

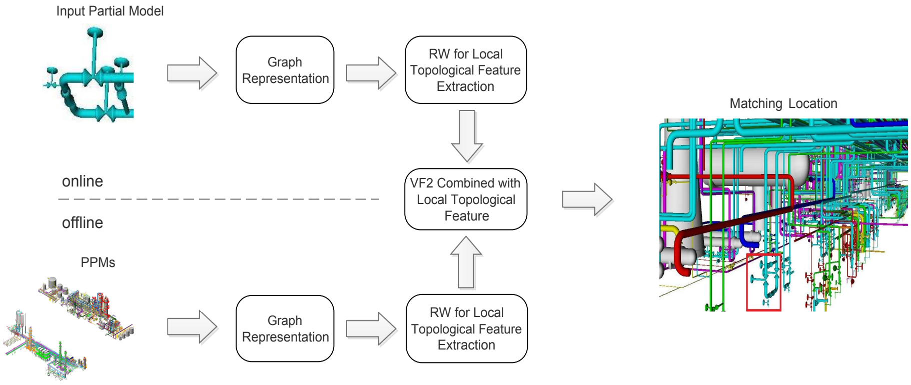

Input Partial Model
Graph Representation
RW for Local Topological Feature Extraction
online
offline
VF2 Combined with Local Topological Feature
Matching Location
PPMs
Graph Representation
RW for Local Topological Feature Extraction

FIGURE 2. Partial matching framework of PPMs. The output is the specific location of the input local structure in the PPM.

In conclusion, few works deal with the partial matching for PPMs. Many previous approaches for 3D&CAD model matching focused on the shape similarity. Corresponding features mainly reflected the shape information. However, distinct from 3D&CAD model matching, partial matching of large scale PPMs should focus on the topology matching. The PPM can be viewed as a large scale assembly model. Only a few works published have addressed the matching problem of assembly models with small component size. Published works addressing matching problem of PPMs do not solve the partial matching problem of 3D PPMs. Therefore, partial matching for large scale PPMs is worthy of in-depth research.

III. PARTIAL MATCHING OF PPMS BASED ON RW ON GRAPHS

A. OVERVIEW OF THE MATCHING FRAMEWORK

The entire matching framework is shown in Fig. 2. In the offline phase, the graph representation of each PPM in the dataset is derived at first. In order to differentiate similar local structures, the local topological feature of each component is extracted based on the RW on the subgraph. In the online phase, similar processing is conducted for the input partial model. VF2 subgraph isomorphism algorithm [41] combined with local topological features is performed for partial matching of PPMs. The output is the specific matching position.

B. GRAPH REPRESENTATION OF THE PPM

Each PPM is defined as an undirected labeled graph. Graphs can be used to model many types of relations and processes between objects [42], [43]. Topological information among components is the core of the PPM. Consequently, we use an undirected labeled graph to represent the PPM saving topology information among components as follow

$$G = (V, E, V_A, E_A) \tag{1}$$

where V is the set of vertices; E is the set of edges; V_A is the set of vertex attributes; E_A is the set of edge attributes.

Each component is represented by a vertex. There is an edge between two vertices if corresponding components are connected directly. The attributes of vertices and edges are defined according to engineering design rules. The attribute of each vertex is defined as a string of numbers representing the type information of corresponding component. Fig. 3 shows the graph representation of a demo PPM.

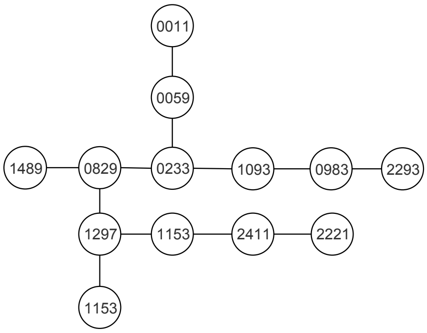


FIGURE 3. The graph representation of a demo PPM. A string of numbers representing the type information of corresponding component is defined as the attribute of the vertex.

TABLE 1. Attributes of some edges shown in Fig. 3.

edges	edge attributes
(0829,1489)	0.231122
(2411,2221)	0.120877
(0983,2293)	0.34745

All the pairwise connected components in the database are statistically obtained, in the form of triple, like $(ComponentA, ComponentB, ConnectedTimes)$. The statistical results reflect some engineering rules to some extent. For example, two valves cannot be connected directly without intermediate components, such as pipes, gaskets, etc. Thus, the connection times between any two valves are 0. Considering this, we use a unique number in $(0, 1)$ to represent one connected pair of components with connected times greater than 0. The numbers are served as the edge attributes in the graph representation. Table. 1 gives some examples of the edge attribute in the graph shown in Fig. 3. With these settings, the graph representation of the PPM can be represented by an adjacency matrix H, where H_{ij} is the attribute of the edge connecting component i and component j.

It is necessary to show the specific matching position in the complex PPM. For this purpose, each component has an ordinal number and coordinates as auxiliary information to display the matching result.

C. LOCAL TOPOLOGICAL FEATURE EXTRACTION USING RW ON LOCAL STRUCTURES

1) RW on graphs

The RW on a graph is a stochastic process where the random walker moves from one vertex to its chosen neighbor according to edge probability or jump directly to any vertex of the graph [8]. Fig. 4 shows an example of the RW on a graph.

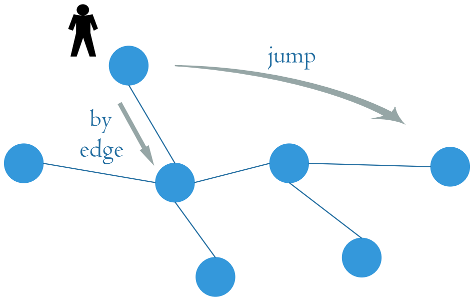


FIGURE 4. Random walk on a graph.

For each transition, suppose the probability of transition by edge is α, and the probabilities of jumping to all vertices are equal. For a graph G with N vertices, the probability that the random walker staying on vertex q at time $t+1$ is updated by

$$\begin{aligned} x_q(t+1) = \alpha \cdot & \sum_{m \in N_q} x_m(t) \cdot p(m,q) \\ & +(1-\alpha)/N \cdot \sum_{n \in G} x_n(t) \end{aligned} \tag{2}$$

where N_q is the set of neighbor vertices of q, $p(m,q)$ is the probability of transition from m to q by edge.

The probabilities that the random walker staying on all vertices at time $t+1$ can be described in a matrix form as

$$\boldsymbol{x}(t+1) = (1-\alpha)/N \cdot \mathbf{1} + \alpha \cdot \boldsymbol{x}(t) \cdot \boldsymbol{P} \tag{3}$$

where $\mathbf{1}$ is a vector where all entries are 1, $\boldsymbol{P}$ is the transition matrix achieved by normalizing the adjacency matrix of the graph G by row.

The steady-state probabilities of the random walker staying on all vertices can be defined as

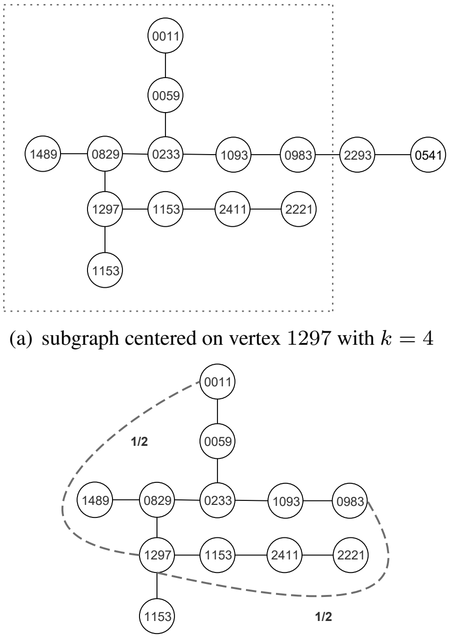
$$\boldsymbol{x}^* = (1-\alpha)/N \cdot \mathbf{1} \cdot (I - \alpha \cdot \boldsymbol{P})^{-1} \tag{4}$$

where I is an identity matrix.

2) Local topological feature extraction based on the RW on augmented subgraph

The RW is performed on an augmented subgraph centered on each vertex which is generated at first. We propose our feature extraction method based on the steady-state probability distribution of the walker's transition on the augmented subgraph.

The augmented subgraph centered on vertex q is generated by adding several virtual edges to the original subgraph. The original subgraph centered on vertex q is represented by $G(q,k)$ that for every vertex $i \in G(q,k)$, the shortest distance from i to q is no more than k. Fig. 5 shows an example of generating an augmented subgraph. A virtual edge is added between center vertex q and vertex p whose shortest distance towards q is k. Suppose the total number of virtual edges in the augmented subgraph is n, the attribute of each virtual edge is $1/n$.



(a) subgraph centered on vertex 1297 with $k = 4$

(b) augmented subgraph centered on vertex 1297

FIGURE 5. The process of generating an augmented subgraph. (a) Extract the local subgraph centered on vertex 1297 with $k = 4$. (b) Add virtual edges between vertex 1297 and the vertices whose distance from 1297 is k. The attribute of each virtual edge is $1/n$, where n is the total number of virtual edges in the augmented subgraph.

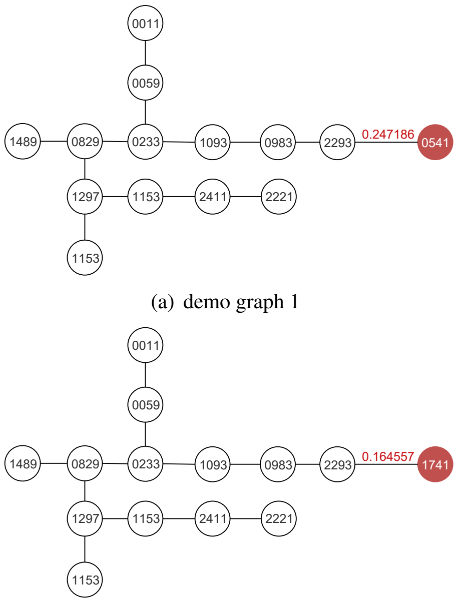
The local topological feature of vertex q is calculated based on the random walk on the augmented subgraph as follow

$$\tilde{\boldsymbol{x}}_{AG(q,k)} = (1-\alpha) \cdot \mathbf{1} \cdot (I - \alpha \cdot \boldsymbol{P}_{AG(q,k)})^{-1} \tag{5}$$

where $AG(q,k)$ is the augmented subgraph centered on vertex q. $\tilde{\boldsymbol{x}}_{AG(q,k)}$ is a vector represents the topological features of all vertices in $AG(q,k)$. The corresponding entry in $\tilde{\boldsymbol{x}}_{AG(q,k)}$, represented by $x^{\star}_{AG(q,k)}$, is the local topological feature of vertex q.

Given the values of α and k, the local topological feature of vertex q is only determined by the local transition matrix $\boldsymbol{P}_{AG(q,k)}$ of the subgraph centered on vertex q. In the offline phase, the local topological features of vertex q can be extracted and stored in a vector form with different k as

$$\boldsymbol{X}_q = [x^{\star}_{AG(q,1)}, x^{\star}_{AG(q,2)}, x^{\star}_{AG(q,3)}, ...] \tag{6}$$



(a) demo graph 1

(b) demo graph 2

FIGURE 6. Two similar demo graphs. The vertex attribute is represented by a string of numbers.

An example is given to verify the effectiveness of the local topological feature for distinguishing similar local structures. Fig. 6 shows two demo graphs. The attribute of each vertex is defined as a string of numbers while edge attributes are defined as mentioned in section III-A. The local topological features of vertex 0829 in two graphs are computed with $\alpha = 0.9$ and $k = 5$. Results are shown in Table. 2. As the local structures centered on vertex 0829 with $k = 5$ in two graphs are different, the local topologicla features are distinct. This property enables us to differentiate similar local structures. Besides, RW on augmented graph has stronger ability to distinguish different local structures.

TABLE 2. Local topological features of vertex 0829 computed with $\alpha = 0.9$ and $k = 5$.

	RW on original subgraph	**RW on augmented subgraph**
0829 (demo graph 1)	0.08261	2.09667
0829 (demo graph 2)	0.08265	2.13016

In conclusion, the local topological feature of each component is extracted as an additional attribute. Thus, each component is represented by a vector. The first entry stands for its type information and the rest are its local topological features.

D. PARTIAL MATCHING OF PPMS BASED ON IMPROVED VF2

As the PPM is defined as an undirected labeled graph, subgraph isomorphism algorithms are feasible solutions for partial matching between PPMs. VF2 algorithm is a widely used depth-first backtracking subgraph isomorphism algorithm. It introduces several feasible rules considering the neighbor vertices of the vertex i that to be matched.

The VF2 algorithm is very suitable for sparse regular graph matching. A component can be connected to a finite number of components in the PPM, such as a pipe can only be connected with two components at two ends. Therefore, the undirected labeled graph representing the PPM can be viewed as a sparse regular graph.

However, the process plant model is quite a large scale assembly model. VF2 algorithm suffers from high time complexity while performing large-scale graph matching. The matching procedure of VF2 is shown in Algorithm 1. The initial candidate pair set $P(s_0)$ is obtained by considering the first vertex n_1 in G_1 and all vertices in G_2. VF2 algorithm is finished when the traversal of $P(s_0)$ is completed. The graph representation of a PPM usually contains thousands of vertices which brings a large size of $P(s_0)$. Besides, computation of feasible rules also takes a lot of time. In order to solve the problems mentioned above, we propose an optimized method of computing $P(s_0)$ based on proposed local topological feature. In the matching process, local topological features is also used for reducing the computation of feasible rules as much as possible.

Algorithm 1 Procedure Match of VF2 Algorithm

```
Input: an intermediate state $s$; the initial state $s_0$ has $M(s_0) = \emptyset$
Output: the mappings between the two graph
  if $M(s)$ covers all nodes of $G_2$ then
    OUTPUT $M(s)$
  else
    compute the set $P(s)$ of the pairs candidate for inclusion in $M(s)$
    FOREACH $p$ in $P(s)$
    if the feasible rules succeed for the inclusion of $p$ in $M(s)$ then
      compute the state $s'$ obtained by adding $p$ to $M(s)$
      call $M(s')$
    end if
    END FOREACH
    Restore data structure
  end if
  END PROCEDURE Match
```

A random component u in the input partial model is selected to generate the optimized $P(s_0)$. Components in the partial model which are used for connecting other components are called port components. If the shortest nodal distance from u towards all port components is l, the selected component will be represented by a two-tuple $[type(u), \tilde{\boldsymbol{x}}_{AG(u,l)}]$. The first entry stands for its type information and the other is its local topological feature.

The size of $P(s_0)$ is reduced by utilizing the local topological feature to generate an optimized $P(s_0)$. If a component m in target PPM have the same two-tuple as

Algorithm 2 Partial Matching between PPMs

```
Input:  a partial process plant model
Output:  specific matching position in the entire PPM
  Generate an optimized $P(s_0)$,the initial state$M(s_0) = \emptyset$
  if  intermediate state $M(s)$ covers all components of the input partial model then
      OUTPUT $M(s)$
  else
      compute the set $P(s)$ of the component pairs candidate for inclusion in $M(s)$
      foreach $p$ in $P(s)$
      if the two-tuples of $p$ are identical then
          compute the state $s'$ obtained by adding $p$ to $M(s)$
          call $M(s')$
      else if the feasible rules for the inclusion of $p$ then
          compute the state $s'$ obtained by adding $p$ to $M(s)$
          call $M(s')$
      end if
      end foreach
  end if
  store data structure and display
```

initial component u which means $[type(m), x^{\star}_{AG(m,l)}] = [type(u), x^{\star}_{AG(u,l)}]$, component pair (u, m) will be added to $P(s_0)$. Considering the actual situation, u is chosen among components that meet the condition $l \geq k$ and the two-tuple is $[type(u), x^{\star}_{AG(u,k)}]$.

The computation of feasible rules is alleviated as much as possible. The two-tuple of one component is a compact representation of its local structure. If the two-tuples of two components are identical, the local structures of these two components are same, which means the feasible rules are satisfied. In this way, the computation of feasible rules is alleviated. Concrete details are shown in Algorithm 2 with underlines.

In conclusion, the local topological feature of each component is extracted based on the RW on the local structure. The VF2 algorithm combined with local topological features is introduced to perform partial matching. A random component is chosen to generate an optimized initial state $P(s_0)$. In addition, the local topological features are used for accelerating the matching process. The matching result is the specific location of the input partial model in PPMs. The whole partial matching algorithm for PPMs is shown in Algorithm 2.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we report the results of experiments to show the effectiveness of our partial matching algorithm for PPMs. All of our experiments are conducted in an Intel quad core 3.7 GHz and 4G memory desktop computer. In section IV-A, we discuss the effect of parameter α on each vertex's local topological feature. Section IV-B gives the partial matching results and comparisons between our improved VF2 algorithm and the original one to show its effectiveness. As shown in Fig. 7, the PPM is a quite complicated model. Therefore, for convenience, we only give the number of vertices in the graph representation of each PPM. The source code will be publicly available at: https://zhiyongsu.github.io.

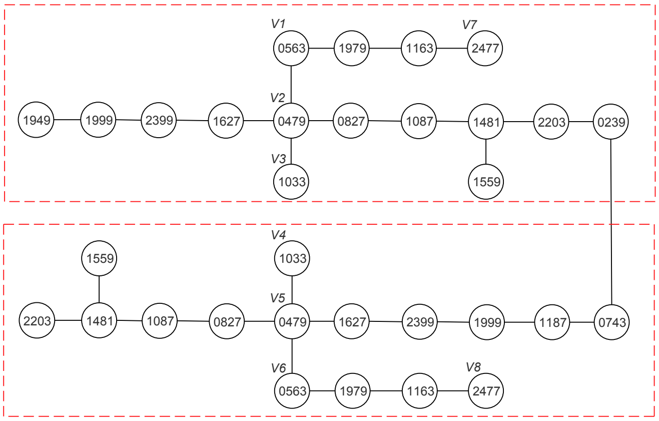


FIGURE 8. A simple graph with similar local structures.

A. ACTION SELECTION PARAMETER α OF THE RW

To evaluate the influence of action selection parameter α on the local topological feature, lots of experiments on different PPMs have been done with $\alpha = [0.5, 0.6, 0.7, 0.8, 0.9]$, respectively. Limited by space, we only give an example here. Fig. 8 is a simple graph containing some similar local structures, e.g., two local structures centered on vertices $V1$ and $V6$ with $k = 6$. Table. 3 shows the topological features of some vertices with similar local structure under different values of α. Fig. 9 shows the distances of local topological feature of two types of components respectively with different α. The results show that a large α is more appropriate for distinguishing similar local structures.

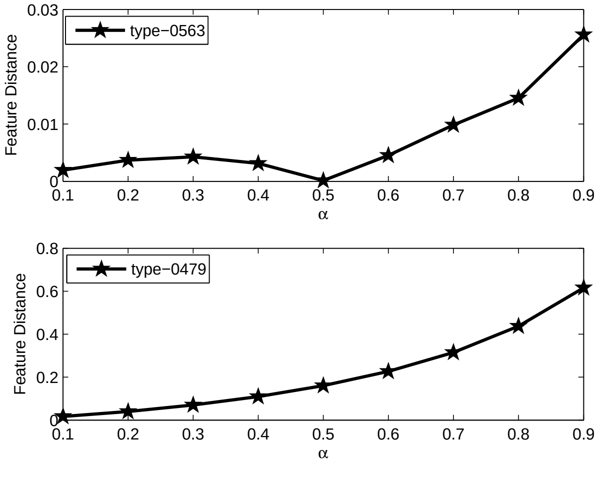
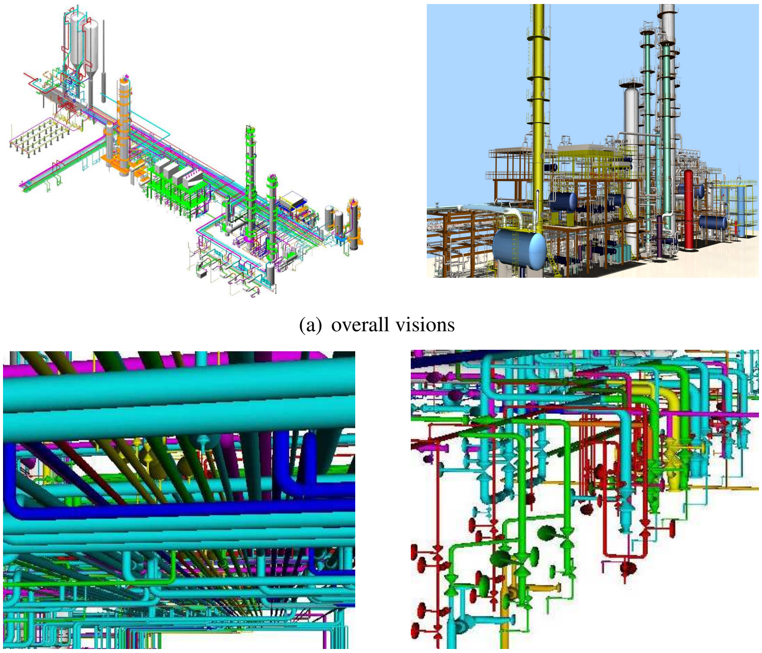


FIGURE 9. Feature Distance of two vertices with same attribute. The upper one is the distance between $V1$ and $V6$. The lower one is the distance between $V2$ and $V5$.



(a) overall visions

(b) local details

FIGURE 7. Examples of the process plant model. a) Overall visions. The PPM usually contains thousands of components of different shapes and sizes. b) Local details. Similar local structures often appear in PPMs.

TABLE 3. Local topological features of some vertices of the graph shown in Fig. 8 with $k = 6$.

α	0563($V1$)	0563($V6$)	0479($V2$)	0479($V5$)	1033($V3$)	1033($V4$)
0.5	1.28556	1.28544	2.17927	2.01874	1.06510	1.06478
0.6	1.40970	1.41422	2.42215	2.19525	1.11534	1.11936
0.7	1.58129	1.59113	2.69675	2.38153	1.18952	1.19738
0.8	1.83623	1.85077	3.03432	2.59709	1.30752	1.31298
0.9	2.26951	2.29510	3.50961	2.89380	1.53977	1.51913

Theoretically, the random walker prefers to jump to any vertex in the subgraph at each transition with a small α. In this condition, the local topology structure has less influence on the transition process. With a larger α, the random walker prefers to transfer by the edge at each transition. Thus, the local topological feature will be more able to reflect the characteristics of the local topology structure. In the extreme condition of $\alpha = 0$, the random walker jumps at each transition. The topological features of all vertices will be same, which makes the feature useless. In other words, the local topological feature of a vertex has a weak discriminating ability of the local structure centered on it when α is small.

B. MATCHING RESULTS AND ANALYTICS

In this section, a group of experiments is conducted to verify the effectiveness of our method. It should be pointed out that there is currently no relevant partial matching algorithm of PPMs that can be used as a comparison. All the local topological features are computed with $\alpha = 0.9$ and $k = 6$.

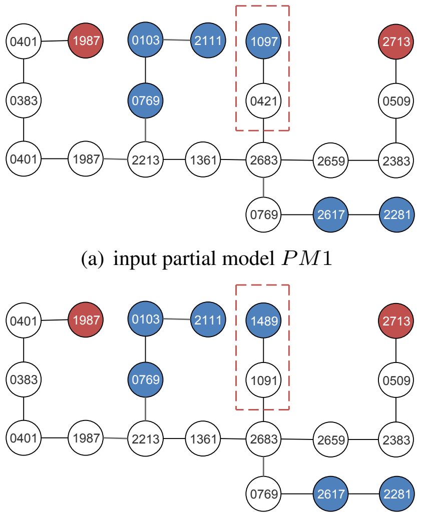
For the first experiment, a small size partial model containing 20 components shown in Fig. 10(a) is designed as the input named $PM1$. The similarities between $PM1$ and three local structures which are defined according to Wen's algorithm [44] are shown in Table. 4. One of them is shown in Fig. 10(b). As mentioned above, the initial component can be selected from vertices marked by blue. The local topological features of some candidate initial components are shown in Table. 5. As shown in the table, whatever the random selected initial component is, it will be able to distinguish these two similar local structures.

TABLE 4. Similarities between similar local structures and $PM1$.

Local Structures	Similarities([0 1])
local structure 1	0.8018
local structure 2	0.8957
local structure 3	0.8434

There are five PPMs named from $P1$ to $P5$ which all contain $PM1$. Besides, each PPM contains several similar local structures as shown in Table. 6. Specific quantities of components in these five PPMs are shown in the brackets in Table 6. The average time of ten runs is shown in Fig. 11.

Since the input partial model is quite a small local structure, only a few components that meet the requirement mentioned in section III-D that can be selected to generate an optimized initial state $P(s_0)$. In this situation, the local



(a) input partial model $PM1$

(b) the similar partial model

FIGURE 10. Two similar local structures. Port components are marked by red. The initial component can be chosen from vertices marked by blue. The difference is in the dashed box.

TABLE 5. Local topological features of some components shown in Fig. 10.

Component type information	Partial model in Fig. 10(a)	Partial model in Fig. 10(b)
2111	2.52500	2.53672
2281	2.48579	2.48671
2617	3.00662	3.00778
0103	2.74016	2.74327
0769	2.10621	2.10716

topological feature of the selected component is able to distinguish similar local structures. In the VF2 algorithm, more similar local structures lead to more matching time as shown in Fig. 11. But the improved VF2 can complete matching more quickly under the interference of similar local structures. The average matching time for these five PPMs is almost the same. In other words, the improved VF2 algorithm is quite efficient.

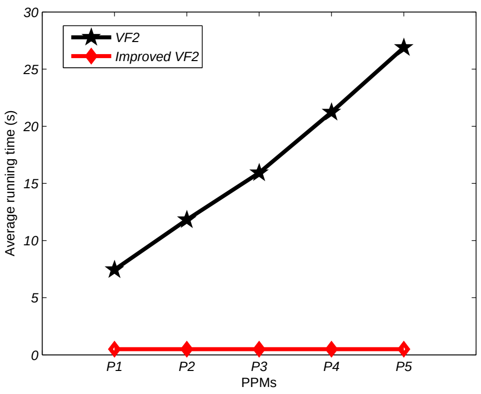


FIGURE 11. Average matching time for $PM1$.

TABLE 6. Similar local structures contained in $P1$ to $P5$.

Local Structures	$P1$ (8037)	$P2$ (8090)	$P3$ (8085)	$P4$ (8013)	$P5$ (8167)
local structure 1	3	6	9	12	16
local structure 2	4	7	8	13	17
local structure 3	4	8	12	14	17

For the second experiment, a partial model containing 93 components is designed as the input named $PM2$. There are three local structures similar to $PM2$. Details are shown in Table. 7. There are three PPMs named from $P6$ to $P8$ which all contain the input local structure $PM2$. Besides, these three PPMs contain a few similar structures defined in Table. 7, respectively. The quantities of all similar local structures contained in each of these three PPMs are shown in Table. 8. The numbers in brackets is the amounts of components contained in each PPM. The average time of ten runs is shown in Fig. 12. There are two curves of the improved VF2 standing for two different selections of the initial component.

TABLE 7. Similarities between similar local structures and $PM2$.

Local Structures	Similarities([0,1])
local structure A	0.732
local structure B	0.7392
local structure C	0.7016

TABLE 8. Similar local structures contained in three PPMs.

Local structures	$P6$(6767)	$P7$(8163)	$P8$(10453)
local structure A	1	1	2
local structure B	1	2	3
local structure C	1	3	3

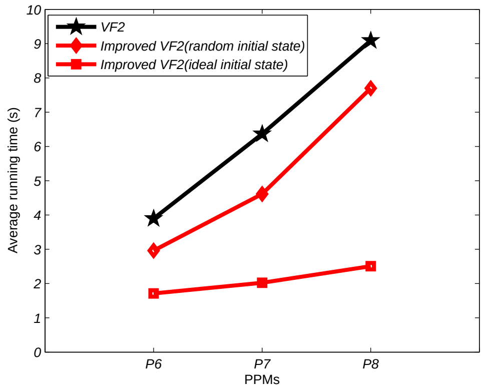


FIGURE 12. Average matching time for $PM2$.

The average matching time is sensitive to the selection of the initial component. The local topological feature is extracted based on the 6-neighborhood structure. When the differences between $PM2$ and its similar local structures are not in the range of 6-neighborhood structure of the selected component, the local topological feature of the initial component cannot distinguish the similar local structures from $PM2$. In this situation, more similar local structures lead to more matching time. If the selection of the initial component is ideal, the matching time remains stable. However, the selection of the initial component is random. There is no information in advance to guide the selection of the initial component.

In conclusion, the time cost of performing VF2 is much influenced by the local structures in the target PPM which is similar to the input partial model. Quantities of components in the five PPMs of the first experiment are almost the same. More similar local structures bring more backtrack times. With the assistance of proposed local topological feature, the improved VF2 can complete the matching process more quickly.

Also, the matching time is influenced by the quantity of components in the input partial model. First, more components lead to more searching and matching time. Second, the quantity of candidate initial components may increase with the quantity of components in the partial model. Sometimes the random selected initial component is not ideal. One solution to guarantee that the selected component is ideal is reducing the quantity of candidate initial components by choosing a larger k. In other words, the value of k can be adjusted according to the input partial model.

In conclusion, while performing partial matching between input partial model and target PPMs containing local structures similar to the input, VF2 combined with the proposed local topological feature can significantly reduce matching time.

C. COMPUTATIONAL COMPLEXITY ANALYSIS

Obviously, the spatial complexity of matching procedure of VF2 is $\mathcal{O}(N)$. The time complexity is $\mathcal{O}(N^2)$ in the best case, and $\mathcal{O}(N!N)$ in the worst case [41]. The computation of feasible rules can be performed in a time proportional to the amount of neighbor vertices of the current candidate pair.

In improved VF2, the computation of feasible rules is mitigated as much as possible. By using proposed local topological feature, the computational burden of feasible rules can be reduced. Besides, the size of $P(s_0)$ in improved VF2 is much smaller by using local topological feature to generate an optimized $P(s_0)$. For example, Table. 9 shows the different sizes of $P(s_0)$ while performing matching between $PM2$ and $P6$. The component in $PM2$ used to generate the initial state $P(s_0)$ is represented by $[2377, 2.61691]$).

TABLE 9. Local topological features of some components shown in Fig. 10.

	size of $P(s_0)$
VF2	6767
using type information(2377)	20
using type information and proposed feature($[2377, 2.61691]$)	1

V. CONCLUSION

In this paper, we propose a partial matching framework for PPMs based on the VF2 algorithm combined with local topological features extracted by RW on graphs. First, the local structure centered on each component is extracted to generate the augmented subgraph representation. In order to distinguish similar local structures, the local topological feature of each component is extracted by using RW on the augmented subgraph representation. The partial matching algorithm of PPMs is based on the VF2 subgraph isomorphism algorithm. In order to alleviate the computing overhead of the VF2 algorithm, the local topological feature is used for generating an optimized initial state. In addition, using local topological features can avoid the computing of feasible rules to a certain extent.

Experimental results show that the proposed method has two advantages:1) proposed local topological features have good discrimination ability for similar local structures; 2) the VF2 algorithm combined with the local topological feature is quite efficient and can meet the needs of industrial applications. The limitations of proposed methods are: 1) models in the dataset requires pretreatment; 2) the selection of the initial component sometimes is not ideal.

In the future, we will focus on the recognition of 2D engineering drawings by the few-shot and one-shot learning method [45]. Furthermore, model retrieval method with 2D engineering drawings as the input is the subsequent research interest.

REFERENCES

[1] N. Ishii, T. Fuchino, and M. Muraki, “Life cycle oriented process synthesis at conceptual planning phase,” *Computers & Chemical Engineering*, vol. 21, pp. S953 – S958, 1997.

[2] R. Wen, W. Tang, and Z. Su, “Topology based 2d engineering drawing and 3d model matching for process plant,” *Graphical Models*, vol. 92, pp. 1 – 15, 2017.

[3] B. Li, Y. Lu, C. Li, A. Godil, T. Schreck, M. Aono, M. Burtscher, Q. Chen, N. K. Chowdhury, B. Fang, H. Fu, T. Furuya, H. Li, J. Liu, H. Johan, R. Kosaka, H. Koyanagi, R. Ohbuchi, A. Tatsuma, Y. Wan, C. Zhang, and C. Zou, “A comparison of 3d shape retrieval methods based on a large-scale benchmark supporting multimodal queries,” *Computer Vision and Image Understanding*, vol. 131, pp. 1 – 27, 2015.

[4] H. Kim, M. Cha, and D. Mun, “Shape distribution-based retrieval of 3d cad models at different levels of detail,” *Multimedia Tools and Applications*, vol. 76, no. 14, pp. 15 867–15 884, 2017.

[5] X. Chen, S. Gao, S. Guo, and J. Bai, “A flexible assembly retrieval approach for model reuse,” *Computer-Aided Design*, vol. 44, no. 6, pp. 554 – 574, 2012.

[6] H. Kim, M. Cha, and D. Mun, “Shape distribution-based approach to comparing 3d cad assembly models,” *Journal of Mechanical Science and Technology*, vol. 31, no. 12, pp. 5627–5638, 2017.

[7] C. Gkantsidis, M. Mihail, and A. Saberi, “Random walks in peer-to-peer networks,” in *IEEE INFOCOM 2004*, vol. 1, 2004, pp. 130–141.

[8] M. Gori, M. Maggini, and L. Sarti, “Exact and approximate graph matching using random walks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1100–1111, 2005.

[9] C. G. Bampis, P. Maragos, and A. C. Bovik, “Graph-driven diffusion and random walk schemes for image segmentation,” *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 35–50, 2017.

[10] E. Ruiz Pujadas, G. Piella, H. M. Kjer, and M. A. González Ballester, “Random walks with statistical shape prior for cochlea and inner ear segmentation in micro-ct images,” *Machine Vision and Applications*, vol. 29, no. 3, pp. 405–414, 2018.

[11] J. Shen, Y. Du, W. Wang, and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1451–1462, 2014.

[12] Y. Liang, J. Shen, X. Dong, H. Sun, and X. Li, "Video supervoxels using partially absorbing random walks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 928–938, 2016.

[13] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1, pp. 107 – 117, 1998.

[14] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3d models with shape distributions," in *Proceedings International Conference on Shape Modeling and Applications*, 2001, pp. 154–166.

[15] M. Ankerst, G. Kastenmuller, H. Kriegel, and T. Seidl, "3D shape histograms for similarity search and classification in spatial databases," in *ADVANCES IN SPATIAL DATABASES*, vol. 1651, 1999, pp. 207–226.

[16] I. Sipiran, B. Bustos, and T. Schreck, "Data-aware 3d partitioning for generic shape retrieval," *Computers & Graphics*, vol. 37, no. 5, pp. 460 – 472, 2013.

[17] K. S. Zou, W. H. Ip, C. H. Wu, Z. Q. Chen, K. L. Yung, and C. Y. Chan, "A novel 3d model retrieval approach using combined shape distribution," *Multimedia Tools and Applications*, vol. 69, no. 3, pp. 799–818, 2014.

[18] Y. Zhou and F. Zeng, "2d compressive sensing and multi-feature fusion for effective 3d shape retrieval," *Information Sciences*, vol. 409-410, pp. 101 – 120, 2017.

[19] L. Luciano and A. B. Hamza, "A global geometric framework for 3d shape retrieval using deep learning," *Computers & Graphics*, vol. 79, pp. 14 – 23, 2019.

[20] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3d shapes," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, 2001, pp. 203–212.

[21] V. Barra and S. Biasotti, "3d shape retrieval using kernels on extended reeb graphs," *Pattern Recognition*, vol. 46, no. 11, pp. 2985 – 2999, 2013.

[22] Y. Sirin and M. F. Demirci, "2d and 3d shape retrieval using skeleton filling rate," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 7823–7848, 2017.

[23] Z. Yasseen, A. Verroust-Blondet, and A. Nasri, "View selection for sketch-based 3d model retrieval using visual part shape description," *The Visual Computer*, vol. 33, no. 5, pp. 565–583, 2017.

[24] Y. Su, W. Li, W. Nie, D. Song, and A. Liu, "Unsupervised feature learning with graph embedding for view-based 3d model retrieval," *IEEE Access*, vol. 7, pp. 95 285–95 296, 2019.

[25] T. Hao, Q. Wang, D. Wu, and J.-S. Sun, "A unified framework for cross-modality 3d model retrieval," *Multimedia Tools and Applications*, vol. 76, no. 19, pp. 20 217–20 230, 2017.

[26] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu, "Multi-view 3d object retrieval with deep embedding network," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5526–5537, 2016.

[27] K. Sfikas, I. Pratikakis, and T. Theoharis, "Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval," *Computers & Graphics*, vol. 71, pp. 208 – 218, 2018.

[28] M. Liu, J. Guo, and Y. Guo, "3d model retrieval and pose estimation for indoor images by simulating scene context," *Graphical Models*, vol. 103, p. 101032, 2019.

[29] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[30] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1626–1633.

[31] P. Li, H. Ma, and A. Ming, "A non-rigid 3d model retrieval method based on scale-invariant heat kernel signature features," *Multimedia Tools and Applications*, vol. 76, no. 7, pp. 10 207–10 230, 2017.

[32] S. E. Naffouti, Y. Fougerolle, I. Aouissaoui, A. Sakly, and F. Mériaudeau, "Heuristic optimization-based wave kernel descriptor for deformable 3d shape matching and retrieval," *Signal, Image and Video Processing*, vol. 12, no. 5, pp. 915–923, 2018.

[33] T. Hong, K. Lee, and S. Kim, "Similarity comparison of mechanical parts to reuse existing designs," *Computer-Aided Design*, vol. 38, no. 9, pp. 973 – 984, 2006.

[34] Z.-M. Huangfu, S.-S. Zhang, and L.-H. Yan, "A method of 3d cad model retrieval based on spatial bag of words," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8145–8173, 2017.

[35] S. Tao, S. Wang, and A. Chen, "3d cad solid model retrieval based on region segmentation," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 103–121, Jan 2017.

[36] A. Cardone, S. K. Gupta, A. Deshmukh, and M. Karnik, "Machining feature-based similarity assessment algorithms for prismatic machined parts," *Computer-Aided Design*, vol. 38, no. 9, pp. 954–972, 2006.

[37] F. W. Qin, S. M. Gao, X. L. Yang, J. Bai, and Q. H. Zhao, "A sketch-based semantic retrieval approach for 3d cad models," *Applied Mathematics-A Journal of Chinese Universities*, vol. 32, no. 1, pp. 27–52, Mar 2017.

[38] A. S. Deshmukh, A. G. Banerjee, S. K. Gupta, and R. D. Sriram, "Content-based assembly search: A step towards assembly reuse," *Computer-Aided Design*, vol. 40, no. 2, pp. 244 – 261, 2008.

[39] K. Lupinetti, F. Giannini, M. Monti, and J.-P. Pernot, "Multi-criteria retrieval of cad assembly models," *Journal of Computational Design and Engineering*, vol. 5, no. 1, pp. 41 – 53, 2018.

[40] M. Rantala, H. NiemistÃű, T. Karhela, S. Sierla, and V. Vyatkin, "Applying graph matching techniques to enhance reuse of plant design information," *Computers in Industry*, vol. 107, pp. 81 – 98, 2019.

[41] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1367–1372, 2004.

[42] S. Poulik and G. Ghorai, "Determination of journeys order based on graph's wiener absolute index with bipolar fuzzy information," *Information Sciences*, vol. 545, pp. 608 – 619, 2021.

[43] S. Poulik and G. Ghorai, "Certain indices of graphs under bipolar fuzzy environment with applications," *Soft Computing*, pp. 5119 – 5131, 2020.

[44] R. Wen, W. Tang, and Z. Su, "Measuring 3d process plant model similarity based on topological relationship distribution," *Computer-Aided Design and Applications*, vol. 14, no. 4, pp. 422–435, 2017.

[45] X. Dong, J. Shen, D. Wu, K. Guo, X. Jin, and F. Porikli, "Quadruplet network with one-shot learning for fast visual object tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3516–3527, 2019.

WEIWEI MAO received the B.S. degree in Automation from the Nanjing University of Science and Technology. He is currently a Ph.D. student of visual computing group in the School of Automation at Nanjing University of Science and Technology.His research interests include model retrieval and machine learning.

ZHUHENG LU received the B.S. degree in Automation from the Nanjing University of Science and Technology.He is currently a Ph.D. student of visual computing group in the School of Automation at Nanjing University of Science and Technology.His research interests include 3D point cloud processing and machine learning.

YUEWEI DAI received the M.S. and Ph.D. degrees from the Nanjing University of Science and Technology, in 1987 and 2002, respectively, all in automation. He is currently a Professor with the School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, China. His research interests are in the areas of information security, signal and image processing.

WEIQING LI is currently an associate professor in the School of Computer Science and Engineering at Nanjing University of Science and Technology, China. He received the B.S. and Ph.D. degrees from the School of Computer Sciences and Engineering, Nanjing University of Science and Technology in 1997 and 2007, respectively. His current interests include computer graphics and virtual reality.

ZHIYONG SU is currently an associate professor of visual computing group in the School of Automation at Nanjing University of Science and Technology, China. He received the B.S. and M.S. degrees from the School of Computer Science and Technology, Nanjing University of Science and Technology in 2004 and 2006, respectively, and received the Ph.D. from the Institute of Computing Technology, Chinese Academy of Sciences in 2009. His current interests include computer graphics, computer vision, augmented reality, and machine learning.