Computers in Industry 116 (2020) 103186

Contents lists available at ScienceDirect

Computers in Industry

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

BIMSeek++: Retrieving BIM components using similarity measurement of attributes

Nanxing Li [a], Qian Li [a], Yu-Shen Liu [a,*], Wenlong Lu [b], Wanqi Wang [b]

[a] School of Software, BNRist, Tsinghua University, Beijing 100084, P. R. China
[b] Institute of Computing Technology, China Academy of Railway Sciences Corporation Limited, Beijing 100081, P. R. China

ARTICLE INFO

Article history:
Received 28 August 2019
Received in revised form 5 December 2019
Accepted 4 January 2020

Keywords:
Information retrieval
Attribute similarity
Domain-specific retrieval
Building information modeling (BIM)
Industry foundation classes (IFC)

ABSTRACT

Building information modeling (BIM) has played a central role in architecture, engineering, and construction (AEC) industry, which also becomes an active research direction in smart buildings and smart cities. With the rapid development and popularization of BIM technology, online BIM resource libraries have grown rapidly. Fast and effective retrieval of BIM components from such great amount of resources has become an urgent demand. Traditional methods such as catalog browsing, keyword matching and shape matching are not capable of delivering satisfactory results, since they cannot extract the domain-specific information carried by BIM components. To resolve the aforementioned issue, we propose a novel similarity measurement and a new retrieval method, and integrate them into the BIMSeek system. The main contributions of our work are as follows. Firstly, we propose a novel algorithm for measuring the similarity between two BIM components based on their attribute information and Tversky similarity. Our proposed algorithm yields the best result in terms of Precision–Recall, F-measure and DCG compared to the traditional Tversky similarity measure and geometry similarity algorithm. Secondly, based on our proposed similarity measurement algorithm, we propose a novel retrieval method of BIM components called query-by-model. We integrate both our proposed similarity measurement algorithm and retrieval method into the BIM retrieval system, named BIMSeek, to greatly improve its retrieving speed and accuracy. Furthermore, we combine the query-by-model and query-by-keyword methods to refine the retrieval results iteratively. Finally, we conduct extensive experiments that compare our proposed method against previous retrieval methods. Results show that our method outperforms previous methods.

1. Introduction

Through recent decades, following the sound, image and video, 3D models have been widely used in the fields of computer graphics, computer-aided design and computer vision as the fourth generation of multimedia resources. The number of shareable 3D models has grown rapidly on the Internet. With advantages such as improving development efficiency, shortening development cycle, and saving development costs, multimedia retrieval technologies has attracted the attention of many researchers [1,2], especially in the field of smart building [3,4] and smart city [5]. Finding a way to enable users to quickly and accurately find 3D models that meet the design requirements and therefore to achieve the reuse of resources has become a hot research topic [6–10].

In the traditional 3D model retrieval field, the descriptor of the model is constructed by extracting the geometric features of the model, and the target models are retrieved by the similarity measure [6]. However, for the 3D models in the field of engineering design, not only the geometric features but also the semantic attributes are included. Therefore, it is not enough to describe the whole model only by extracting geometric features. 3D model retrieval based on the content of the model itself can better support the retrieval and reuse of 3D models in the engineering field, and thus becomes a research topic. In this work, we mainly focus on the retrieval of BIM (building information modeling) components.

In the field of architectural engineering, BIM is an engineering data model based on 3D digital technology that contains various relevant information in construction projects [11]. BIM components not only contain the geometric information of the building components (such as length, width, depth, etc.), but also the seman-

* Corresponding author at: School of Software, BNRist, Tsinghua University, Beijing 100084, P. R. China.
E-mail address: liuyushen@tsinghua.edu.cn (Y.-S. Liu).
URL: http://cgcad.thss.tsinghua.edu.cn/liuyushen/ (Y.-S. Liu).

https://doi.org/10.1016/j.compind.2020.103186
0166-3615/

tic information of the building components (such as materials, manufacturers, fireproof rate, etc.), in addition to the association information between the components (such as a door being embedded in a wall). It can be concluded that with the help of BIM components, data sharing throughout a building's life cycle can be better achieved. The IFC (industry foundation classes) international standard defined by the IAI (International Alliance for Interoperability) organization is the most utilized data exchange standard of BIM [12]. Therefore, in our work, we use IFC files to represent BIM components and conduct our research on the retrieval of BIM components.

With the rapid rise of BIM technology in the AEC (Architecture, Engineering and Construction) field, a large number of BIM resource libraries have emerged on the Internet. Popular ones include BIMobject,[1] Autodesk Seek,[2] Arcat,[3] the NBS National BIM Library,[4] 3D Warehouse,[5] Modlar, SmartBIM, and many more. These libraries usually have thousands to tens of thousands of BIM components. For example, BIMobject has more than 270 brands, more than 4000 products and more than 1.9 million downloadable files. Faced with such an ever-larger library of 3D models, designers need to shift their focus from how to build 3D models from scratch to how to build new models that fit the needs based on existing models. Gunn [13] stated that 40% of components can be redesigned on existing models, 40% of components can be modified through existing models, and only 20% of components need to be redesigned. Ullman [14] believed that more than 75% of designs can reuse previous designs to meet new demands. It can be seen that the demand for component reuse is quite large. Therefore, how to quickly and accurately find the components that meet the designer's needs becomes a key issue. Leizerowicz [15] found out in a survey that designers usually spend 60% of their time searching for components. Thus, providing a retrieval system that can quickly and accurately find 3D models, allowing designers to retrieving the required model in the library with a wide variety of components is extremely important to the designer.

Catalog browsing, keyword matching, and geometry matching are the mainstream retrieval methods in current BIM component retrieval systems. However, these methods lack the extraction of domain-specific information carried by the BIM components. For example, catalog browsing is very time-consuming, and since the catalogues are all labeled by humans, it lacks a unified standard, which makes it difficult keep its consistency with requirements expressed by users. Keyword matching is difficult to meet the complex needs of users if there is no detailed and complemental textual descriptions associated with the BIM components. Geometric shape similarity matching only considers geometric information, and different forms of the same model will be considered different. Therefore, users cannot quickly and accurately find a component that satisfies the domain-specific demand.

The contributions of our work are as follows: We first propose a similarity measurement of BIM components based on attribute information and Tversky similarity [16], which combines the domain-specific information provided by BIM models with the classic Tversky similarity. We designed our similarity measurement based on the information content theory proposed by Resnik [17], which allows us to fully utilize the information carried by BIM components. Based on this similarity measurement, a new retrieval method named query-by-model is proposed. That is, the similar BIM components are retrieved by uploading an inquiry BIM component. We integrate our proposed retrieval method into the BIMSeek retrieval system [18,19], which is a semantic-based retrieval system for BIM documents. Together with our proposed retrieval method and the BIMSeek system, users can also retrieve the components by keyword and retrieve similar components to continuously optimize the search results. This type of retrieval is not yet widely used in the mainstream BIM search systems on the Internet. With our proposed query-by-model method and the keyword retrieval and catalog browsing features of the original BIMSeek system, we are able to provide to the users with richer ways to retrieve BIM components. We built a BIM component library of a total of 10,000 components from Arcat, Autodesk Seek and BIMobject to conduct our experiments on BIM components retrieval. It can be seen from our experiments that the information and Tversky-based BIM component attribute similarity measure method can match the similar models more accurately. Therefore, the BIMSeek retrieval system can more accurately retrieve the models that match user needs.

1.1. Related work

Studies on BIM retrieval can be summarized in three categories: keyword-based retrieval methods, content-based retrieval methods and semantic-based retrieval methods.

1.1.1. Keyword-based retrieval methods

The keyword-based retrieval methods use various algorithms to match keywords in a document after getting keywords input by the user [20]. Commonly used algorithms are like Boolean, probabilistic and vector space models [21].

A Boolean model means that each document in the database is treated as a series of index words. The search statements are expressed with Boolean expressions and only documents with an exact match are retrieved. This results in too many or too few results being retrieved and partial matching is not supported. In addition, the Boolean model not being able to provide a quantitative ordering of search results, and it is difficult for Boolean expressions to fully and accurately describe user intent. Probability models use probability theory to solve retrieval problems. The probabilistic model assumes that there is an imaginary set of ideal answer documents, and the similarity calculation is based on the estimating the probability of the query term belonging to this imaginary set, which is calculated based on Bayesian theory [22]. Thus, longer query term and relevant documents are required for a more accurate result. The vector space model can solve the shortcomings of the Boolean model and the probability model. It is based on the assumption that if two documents are similar, they should have as many similar indexing words as possible. By assigning weights to query terms and document index words, local matching between query words and documents can be achieved [23]. The advantage of the vector space model is that it can perform local matching, allowing fuzzy query and sort the results of the query [24]. However, the vector space model assumes that the index words are independent of each other, which is its disadvantage.

1.1.2. Content-based retrieval methods

The content-based retrieval methods extract the feature information of a 3D model and retrieve the similar 3D models by similarity matching. The intrinsic feature information of 3D model includes geometric shapes, colors, materials and the like.

There are many retrieval methods for 3D CAD models [6], which can be roughly divided into four categories: retrieval based on shape matching, retrieval based on topological structure, retrieval based on image comparison, and retrieval based on functional description. Retrieval based on shape matching mainly extracts the shape feature of a 3D model and uses it as a basis for searching [7,8,25–27]. The drawback of this search method is that different

[1] http://bimobject.com.
[2] Autodesk Seek's BIM content is now hosted on the BIMobject.
[3] https://www.arcat.com/.
[4] http://www.nationalbimlibrary.com.
[5] https://3dwarehouse.sketchup.com.

forms of the same object are usually considered different. For example, a door that's open or closed, or opened at different angles can be considered the same door in different forms. Although it is the same object, their 3D shape features are largely different. Retrieval based on the topology is mainly to compare the topological structure of the 3D model [28]. If the topological structures of two models are similar, they are considered to be similar [29]. This solves the issue that shape matching has for different forms of the same object. But this approach is more likely to falsely consider different for similar models. For retrieval based on image comparison, the 3D model is converted into a set of 2D views/images. Since the 2D image retrieval technology is very mature, the 2D model retrieval technology can be used to retrieve the 3D model [7,8,30–33]. However, the above-mentioned three kinds of methods do not consider the domain-specific information of 3D models. Unlike the ordinary 3D models, a CAD model generally has a specific manufacturing, design and application environment, so the manufacturing, design, function and other semantic attributes carried by the 3D CAD model can be used for retrieval, such information are referred as domain-specific information. This is also the case with the retrieval method based on the functional description of products, where the similarities of product information between the 3D models are utilized for the retrieval of the model [34,35].

1.1.3. Semantic-based retrieval methods

Semantic-based retrieval methods usually includes three parts: obtaining semantic knowledge, expressing semantic knowledge, and matching and learning semantic knowledge [36]. Different from geometric matching, semantic retrieval pays more attention to the extraction and use of domain knowledge related to the model, and the domain knowledge is often a textual description describing the function and characteristics of the model from the aspect of its design parameters [37]. The easiest way to get semantic knowledge is to manually tag the 3D model. However, this method is labor intensive and the definition is likely to be non-standard. Therefore, automatic semantic annotation has emerged. By constructing a text semantic library, the semantics of the text are mapped to the model, and the components are semantically retrieved [38,19]. Domain ontology is often used to represent semantic knowledge. Domain ontology refers to a set of concepts (or terms) used to describe or express knowledge of a particular domain and their relationship to each other [39,18]. Matching and learning semantic knowledge improve the retrieval efficiency by establishing domain knowledge and adaptively adjustment through related feedback, active learning and other techniques after establishing the mapping between 3D model and domain knowledge [36,40,9] (Table 1).

BIMSeek [18,19] is a retrieval system for BIM components that utilizes semantic-based retrieval methods. In this paper, we improve upon the BIMSeek system with our proposed retrieval method, further improving its retrieval performance.

In view of the fact that the BIM component in the AEC field itself contains a lot of domain-specific information, such as the material of the building component, fireproof rate, production information, etc., we use the content-based retrieval method to retrieve the BIM component itself by extracting the attribute information.

2. Method

In this section, we introduce the proposed method, named *BIMSeek++*, to retrieve BIM components using their attribute similarity measurement. Firstly, we propose a similarity measurement between two BIM components based on attribute information and Tversky similarity. Then, based on the proposed measurement, a query-by-model BIM component retrieval method is developed. That is, similar BIM components are retrieved by uploading a query BIM component. We also integrate this retrieval method into the BIM document retrieval system BIMSeek [18,19] and combined it with keyword search capability for further refinement of search results. For retrieval systems, the speed of retrieval is critical. Therefore, in order to improve the retrieval speed, file cache is added to the BIMSeek retrieval system.

Fig. 1 shows the flowchart of our BIMSeek++ method. After a query model is uploaded, the system first checks whether or not it is stored in the file cache. If so, the system will return the saved results; otherwise, it will go through the retrieval process. Both our proposed similarity measurement and query-by-model method are utilized for the retrieval process.

2.1. Building of a BIM component library

Before performing a BIM component retrieval, it is first necessary to build a library of BIM components, which should contain a sufficient number of BIM components with various types to ensure that a good enough result can be obtained when uploading a BIM component. In addition, the BIM components in the repository should have a unified storage format. Our BIM components are acquired from various mainstream online BIM component libraries, which will be detailed in the experiment section. As an open and neutral data format specification for BIM, industry foundation classes (IFC) [41] plays a crucial role to facilitate interoperability between various software platforms. The IFC data format has been widely supported by the market-leading BIM software vendors. Many recent studies also demonstrate the IFC viability in various applications [18,19,42–44,40]. Therefore, in this paper, we use the IFC standard to represent BIM components, which will be processed into corresponding attribute vectors in the pre-processing stage. The pre-processing is performed in the following four steps, as shown in Fig. 2.

2.1.1. Step 1: Attribute extraction

There are 653 entities defined in IFC2X4, and each IFC file represents a component. In general, small components have hundreds or thousands of fields, and large components may have tens of thousands or even hundreds of thousands of fields. However, not all field information are necessary, only the field information that can describe the components' characteristics. The attributes of components can be divided into three categories [45]: geometric attributes, semantic attributes and relationship attributes.

- *Geometric attributes*: using geometric information such as numbers to describe the properties of a component, such as width, length, depth, and so on.
- *Semantic attributes*: using semantic information such as textual natural language to describe the attributes of components, such as materials, colors, types, and so on.
- *Relationship attributes*: describing the relationship between one component and another ones, such as the inclusion relationship of a door being embedded in the wall.

Since our research is concerned with only the attributes of the component itself, therefore, we focus mainly on the geometric attributes and semantic attributes, without considering the relationship attributes with other components. In an IFC file, *IfcPropertySet* describes the attribute information set of a component, and associates with the component through *IfcRelAssociates*. Each *IfcPropertySet* contains one or many *IfcProperty*. Attributes are divided into simple attributes and complex attributes. Complex attributes are combination of simple attributes. Therefore,

Table 1
A summary of BIM model retrieval methods.

Category	Model	Description	Characteristics
Keyword-based	Boolean model	Search statements are expressed with Boolean expressions.	Difficult for Boolean expressions to fully and accurately describe user intent.
	Probability model	Utilizes Bayesian theory to estimate probability.	Requires longer query term and relevant documents.
	Vector model	Represents documents as vectors of index words.	Assuming index words are independent of each other affects accuracy.
Content-based	Shape matching	Utilizes similarity based on 3D shape feature	Consider different forms of the same object as different.
	Topological structure	Calculate similarities based on topological structure.	Similar objects might have different topological structure, which results in inaccuracy.
	Image comparison	Utilizes 2D image retrieval methods after transforming 3D models into 2D.	The transformation from 3D to 2D affects the result greatly.
	Functional description	Utilizes similarity of functional description between models for retrieval.	Utilizes the domain-specific knowledge for CAD models.
Semantic-based		Calculates distance between models based on the likeness of semantic content.	Able to handle the syntax variations and semantic complexities in BIM models.

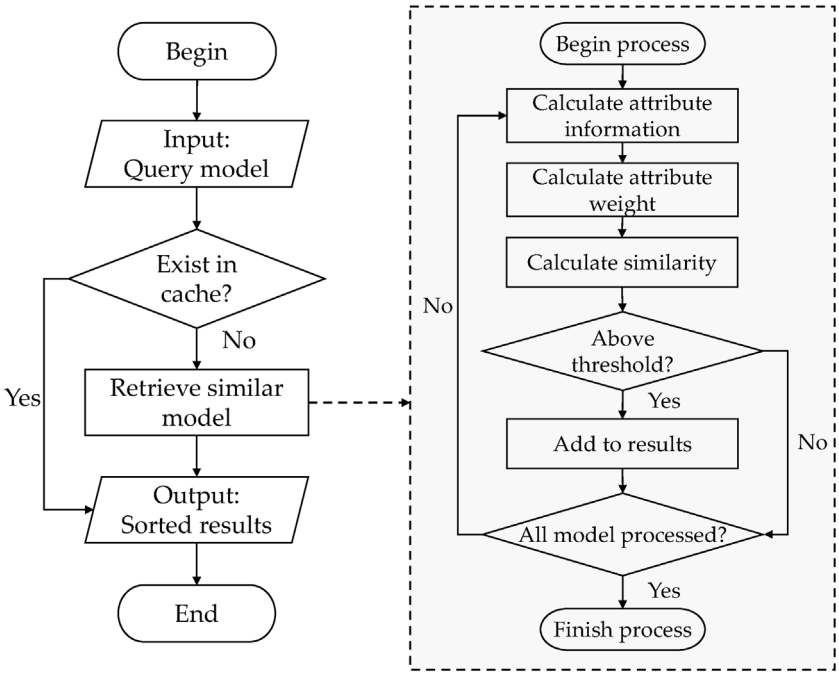

Fig. 1. The flowchart of our BIMSeek method.

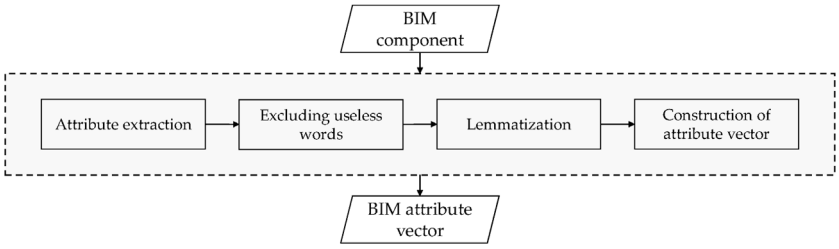

Fig. 2. Flowchart of the preprocessing process.

the six simple attributes, *IfcPropertySingleValue*, *IfcPropertyEnumeratedValue*, *IfcPropertyBoundValue*, *IfcPropertyTableValue*, *IfcPropertyReferenceValue*, and *IfcPropertyListValue*, can cover all geometric and semantic attributes. By extracting all information of the simple attributes, we can obtain all attribute information of a given component, see Table 2. A total of 17,219 different attributes are utilized in our experiments, with an average of 64.2 attributes for each component.

Table 2
Example of attributes utilized.

Category	Geometric attributes	Semantic attributes	Relation attributes
Wall	Length	Type	Connected to
	Height	Curvature	Has openings
	Thickness	Fire rating	Decomposes into
Window	Window width	Type	Connected to
	Screen width	Screen material	Has openings
	Sill width	Has WINDOW screen	Has coverings

Table 3
List of excluded words.

http://, reference, description, project, title, manufacturer, copyright, name, drawn by, schedule, years, months, revision, numberofstoreys, date, url

Step 2: Excluding useless words

In Step 1, all attribute information of the component is extracted. However, not all attribute information is useful. Some information, such as the material information, height, width, etc. of the component, plays an important role in reflecting the characteristics of the component and needs to be extracted. While there is some information such as version information, the project information has basically no effect on reflecting the characteristics of the component and needs to be excluded. For example, "Project Address", "Project Number", "Manufactory", "Copyright", "Tile", "Revision" and other attributes. We manually modified the list of removed words, as shown in Table 3.

2.1.2. Step 3: Lemmatization

Since the attribute names and attribute values defined in the IFC attribute field are artificial, different people may express the same attribute differently. Words with the same root and words that are synonymous should be treated as the same word. In our implementation, WordNet [46], an English lexical database, is utilized for synonym categorization. This avoids the fact that the same attributes might be considered different when the similarity is measured, thereby improving the accuracy.

2.1.3. Step 4: Construction of attribute vector

A vector model is using a vector consisting of a series of indexed words to represent a document. Through the above three steps, the key attributes of the component have been obtained, and these attributes are grouped into a vector, each of which consists of two parts: the attribute name and the attribute value, which is like key-value pair data storages for attributes.

2.2. Similarity measurement based on Tversky similarity

The most critical issue in retrieval is the measurement of similarity between two objects. We use IFC files to represent BIM components. Since the IFC file contains all the geometric and semantic attributes of the BIM component, each BIM component uses a corresponding attribute vector representation, so that the similarity measure of the BIM component is converted to the similarity measure of the component attribute vector.

In our work, we combine the semantic similarity of information content and the semantic similarity based on feature attributes. Resnik [17]'s information calculation formula is a classic algorithm for calculating semantic similarity of information content, while the Tversky similarity model proposed by Tversky [16] is a classic algorithm for calculating semantic similarity based on feature attributes. We combine the two formulas and incorporate the attribute information of the BIM component. Our proposed method first calculates the attribute information of the BIM component, then combines the information of each attribute with a corresponding weight value, which will be detailed in Section 2.2.2, and finally integrates it into the Tversky similarity model to calculate the similarity between two given BIM components.

2.2.1. Attribute information content of BIM components

The calculation formula of *information content (IC)*, as proposed by Resnik, can be denoted as:

$$IC(c) = -\log p(c). \tag{1}$$

The quantized amount of information content is calculated by the probability that the concept appears in a document set. $p(c)$ refers to the probability that concept c appears in the document set. Taking the negative log value of $p(c)$ indicates that the greater the probability that concept c appears, the smaller the amount of information it represents. Based on the information quantity calculation formula proposed by Resnik and the semantic attributes of BIM components, we define the *attribute information content* of BIM components by

$$\begin{aligned} IC(pname\_pvalue) &= -\log p(pname\_pvalue) \\ &= -\log \frac{count(pname\_pvalue)}{total\_bims}. \end{aligned} \tag{2}$$

The attribute name and attribute value of each component are regarded as a whole (hereinafter referred to as attribute). Since each attribute appears only once in a BIM component, $count(pname\_pvalue)$ represents in how many components the attribute $pname\_pvalue$ has appeared. $total\_bims$ refers to the total number of BIM components. This formula indicates that the greater the probability of occurrence of an attribute, the smaller the amount of information. In our experiments, we calculate the information content on the scale of the whole dataset. We save the attribute information of all BIM components to a computer-readable intermediate document for easy usage in the following steps.

2.2.2. Attribute weight of BIM components

Each BIM component is processed into an attribute vector. The attribute vector contains both geometric attributes (length, width, etc.) as well as semantic attributes (materials, vendors, etc.). The weight value of the attribute is set to 1 by default. When the attribute names are the same, in order to ensure more accuracy in the calculation of similarity, the weight value of the attribute is calculated in the following two cases:

Case 1: Geometric attributes. In practice, there may be minor differences between two instances of the same type of components. Therefore, an acceptable margin of difference in geometric attributes should be allowed. In this paper, a threshold of 5% is set.

Attributes with a difference ratio greater than 5% is treated as different attributes, and its weight value is set to 0; while attributes with the difference ratio less than 5% is considered to be the same attribute, but its weight value will be scaled down accordingly. For example, two BIM components A, B, the length of component A is 200 cm, and the length of component B is 190 cm, the difference is less than 5%, thus they are considered as the same attribute, and the weight is set to 190/200 = 0.95.

Case 2: Semantic attributes. As an example of two BIM components A and B. If some but not all words in a property in component B can be matched with component A, for example, component A has "**Metal-Aluminum-Eastern-Andized-Bronze**" and component B has "**Metal-Aluminum-Eastern-Anodized**", we still consider that the two properties are the same, but the weight is correspondingly reduced. In our implementation, the weight is set to $W(B\_to\_A) = n(shared\_words)/n(total\_words\_in\_A)$. Therefore in this example, the weight of B to A is set to 4/5 =0.8, and A to B is set to 4/4 =1.

But if the two attribute values are completely different, then the weight value is 0.

In addition, WordNet [46] is utilized to calculate similarity in the case of custom properties, where different people might uses different words to express the same meaning.

2.2.3. Attribute similarity measurement based on information content and Tversky similarity

In this section, based on Sections 2.2.1 and 2.2.2, the formula of Tversky similarity calculation based on attribute information quan-

tity is proposed, and the similarity between two given components is calculated.

The attribute similarity calculation formula proposed by Tversky can be denoted as:

$$\mathrm{sim}(A, B) = \frac{\alpha|A \cap B|}{\alpha|A \cap B| + \beta|A/B| + \lambda|B/A|}, \tag{3}$$

where A and B represent the attribute vectors of two BIM components, respectively. According to the Tversky similarity calculation formula, the formula only counts the number of features in the intersection and difference set of the two components A and B, and combines them into a formula to reflect the similarity between A and B. It does not consider the attribute information of A and B in the formula.

In order to make the Tversky formula consider the attribute information content of BIM components, we combine it with the attribute information content of the BIM components and its attribute weight values, and propose the similarity measurement formula of BIM components, denoted as:

$$\mathrm{sim}\,(A, B) = \frac{\alpha f\,(I_A \cap I_B)}{\alpha f\,(I_A \cap I_B) + \beta f\left(I_A/I_B\right) + \lambda f\left(I_B/I_A\right)}, \tag{4}$$

where I_A and I_B represent the attribute vectors of component A and component B, respectively. α, β and λ are the blending factors, which are denoted as:

$$\begin{cases} \alpha = \dfrac{\sum_{k \in (I_A \cap I_B)} IC(k)}{M} \\ \beta = \dfrac{\sum_{k \in (I_A / I_B)} IC(k)}{M} \\ \lambda = \dfrac{\sum_{k \in (I_B / I_A)} IC(k)}{M}, \end{cases} \tag{5}$$

$$M = \sum_{k \in I_A \cup I_B} IC(k), \tag{6}$$

where $IC(k)$ is the attribute information content of k, as computed using Eq. (2). M represents the sum of the information content of the union attributes of component A and component B.

It should be noted that the function $f(I)$ in Eq. (4) represents the sum of the information content of all attributes in the set multiplies the corresponding weight values, denoted as:

$$f(I) = \sum_{i=1}^{n} IC(i) \times W_i \tag{7}$$

Assuming that the number of attributes in I is n, $IC(i)$ represents the information amount of the ith attribute, and W_i represents the weight value of the ith attribute. Through the information content and Tversky BIM component attribute similarity measurement formula, the similarity between any two BIM components can be calculated.

2.3. Retrieval method based on model query

Based on the similarity measure algorithm of BIM components, we propose a BIM component retrieval method based on model query and applies it to BIMSeek retrieval system. That is, by uploading a BIM component, a model component similar to it is retrieved. This type of retrieval is necessary in situations where user requirements are complex and have component sketches or prototypes on hand. In addition, this approach, combined with keyword retrieval, allows the user to narrow the scope of search results iteratively.

In the retrieval system, the retrieval speed is very important. Therefore, we add the cache system to the retrieval system. In addition, in order to improve the speed of reading cache files, the algorithm uses a hash table to store the component result information of the relevant retrieval.

Table 4
Source of BIM components used in our experiments.

Library	# of components crawled
Autodesk seek	4000
BIMobject	4000
Arcat	2000

2.4. BIMSeek system

With our proposed similarity measurement and retrieving method, we improve the BIM component retrieval system BIMSeek [18]. With our proposed query-by-model feature, the upgraded BIMSeek system provides four basic functions: keyword search, advanced search, model query, and catalog browsing, as shown in Fig. 3. By integrating our proposed query-by-model method, our system is capable of retrieving with an user uploaded model (Fig. 4), or retrieve similar models with a keyword search result model (see Fig. 5).

The BIMSeek system consists of a three-layer architecture of user interface layer, data processing layer and data storage layer. The architecture diagram of the model query retrieval system is shown in Fig. 6. There are two ways for model query retrieval. The first is that the user uploads a model to find a similar model. The second is that the user uses the keyword search to find a model, then click on the model's "find similar" button to find all models similar to the model. Therefore, the user interface layer is mainly to transmit the user uploaded or selected model to the BIMSeek retrieval system. When the retrieval system obtains the sorted retrieval result, it is output to the user. The data processing layer is for obtaining the corresponding result in the cache file or runs the model retrieval algorithm to retrieve in the BIM library for BIM components similar to the uploaded BIM component. The data storage layer mainly stores the pre-processed crawled BIM components.

The entire query system can be seen as two parts, the processing of BIM components and the retrieving of BIM components. The processing involves the crawling and pre-processing of BIM components, mainly to prepare a sufficient number of BIM components of a variety of classes for components; the retrieving involves file caching and retrieval algorithms, and the process of similarity matching.

3. Experiments

3.1. Acquiring of BIM components

In recent years, with the rapid development of BIM, a large number of BIM resource libraries have emerged on the Internet. At present, the main mainstream are BIMobject, Autodesk Seek, Arcat, SmartBIM, Modlar, National BIM Library, and so on.

BIM component models used in this article is mainly crawled from three major websites, as shown in Table 4.

Online BIM libraries such as Autodesk Seek, BIMobject and Arcat all have their own categorization. Since the BIM components used in our experiments come from these libraries, we use their own categorization as benchmark. A total of 23 categories are used in our experiments, as listed in Table 5.

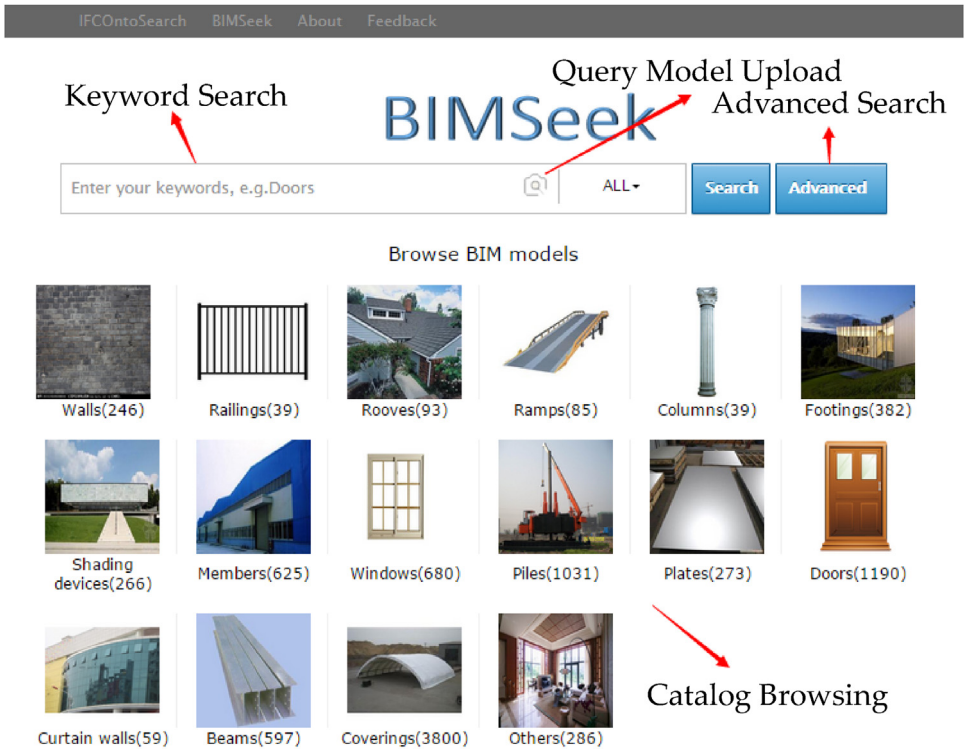

Fig. 3. Features of the upgraded BIMSeek system.

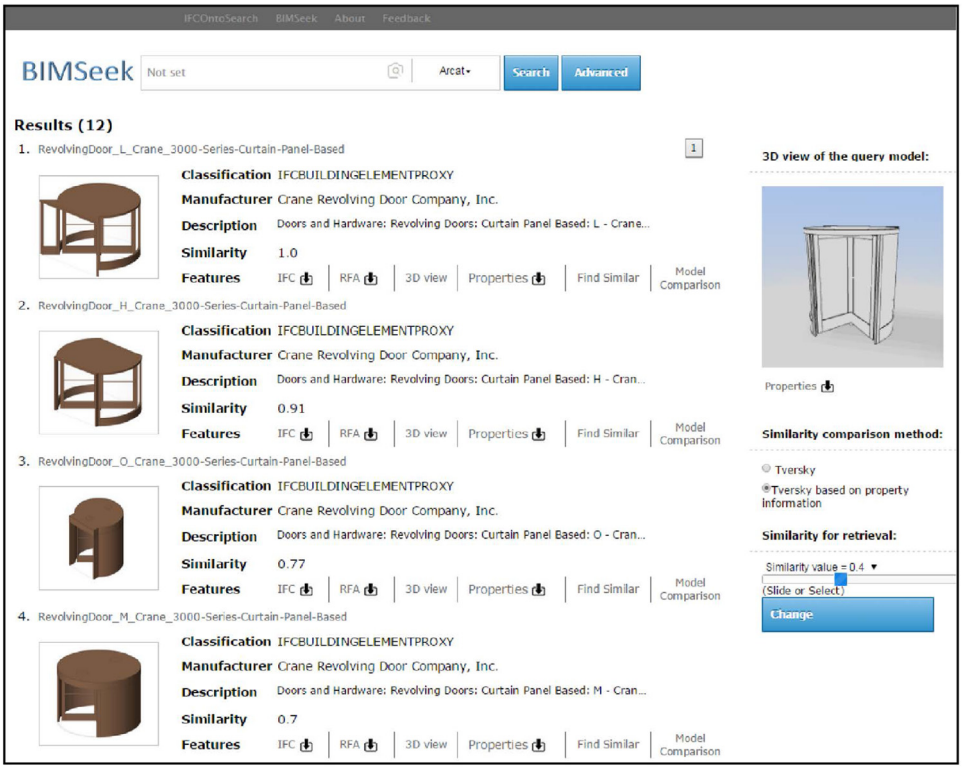

Fig. 4. Example of model query results.

3.2. Evaluation metrics

We use Precision–Recall (PR), F-measure and discounted cumulative gain (DCG) as evaluation methods in our experiments.

3.2.1. Precision–Recall

In retrieval systems, PR curves are commonly used to evaluate the performance of search results. Precision represents the proportion of the truly relevant part of the retrieved results, while recall

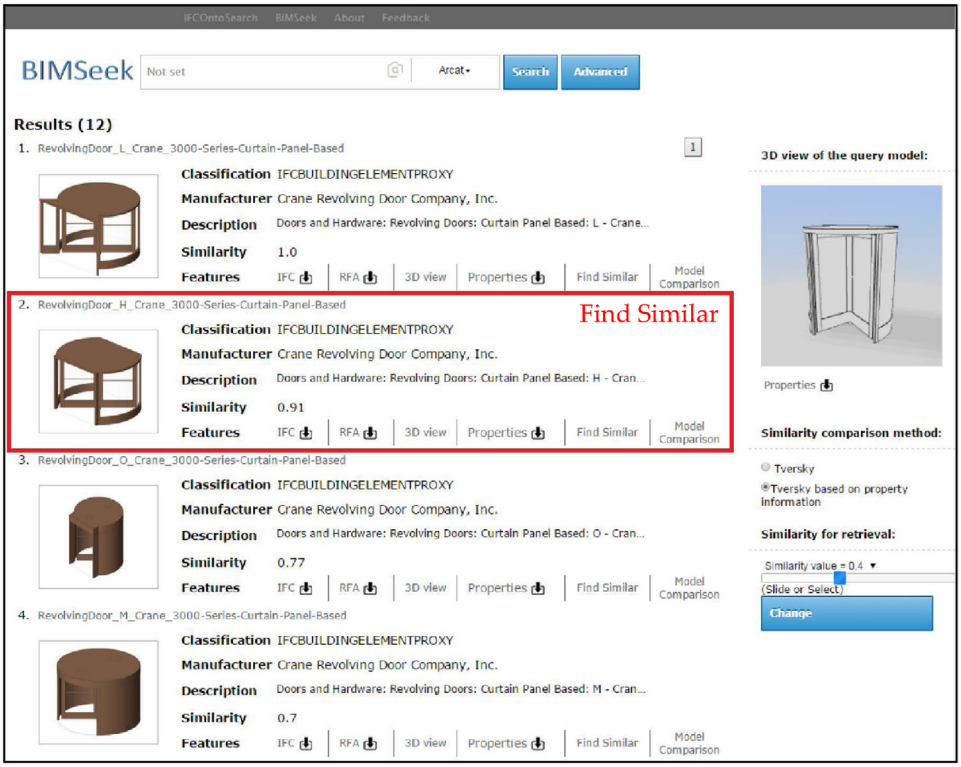

Fig. 5. Example of model query results.

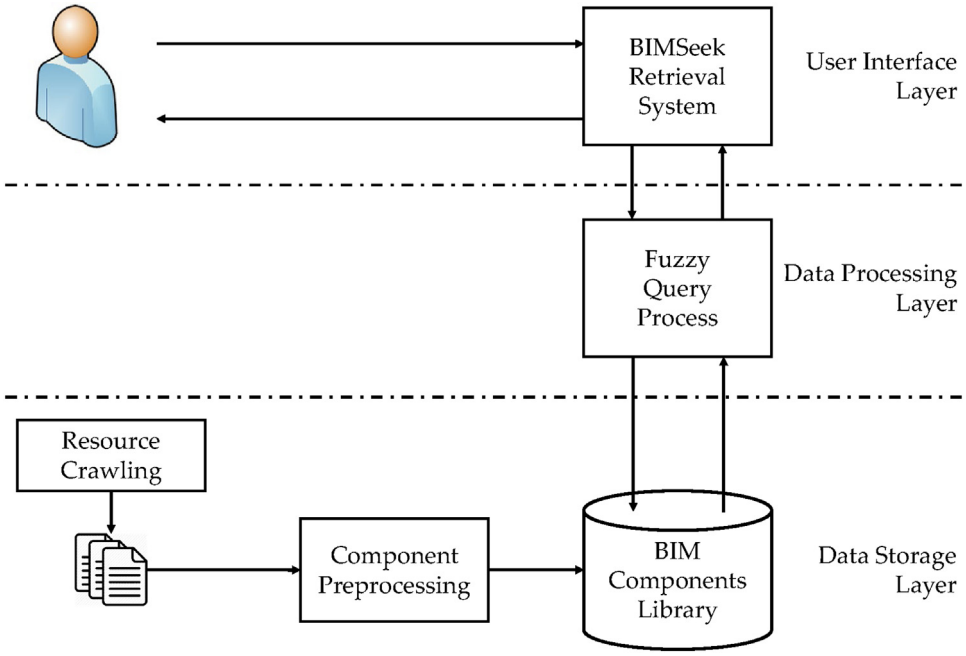

Fig. 6. Architecture diagram of our retrieval system.

represents the proportion of the parts retrieved in the real relevant set:

$$P = TP/(TP + FP) \times 100\%, \quad (8)$$

$$R = TP/(TP + FN) \times 100\%. \quad (9)$$

TP, FP, FN and TN are defined as in Table 6.

Ideally, the more relevant models in the database are retrieved, the better, that is, the greater the recall rate, the better. At the same time, in the retrieved models, the more relevant, and the less irrelevant, the better. Therefore, the high precision and recall rate are ideal, because it means that most of the retrieved results are relevant, and most of the relevant ones are retrieved, But this is very difficult to achieve. A more convex PR curve means higher precision and recall rates are achieved at the same time. Thus, generally, the more convex the PR curve is, and the closer to 1 when the value of Recall is 0.1, the better the retrieval performance.

Table 5
List of component categories.

Generic Models, Walls, Mechanical Equipment, Air Terminals, Speciality Equipment, Plumbing Fixtures, Casework, Structural Framing, Lighting Fixtures, Doors, Curtain Panels, Furniture, Furniture Systems, Electrical Equipment, Windows, Duct Accessories, Fire Alarm Devices, Pipe Accessories, Pipe Fittings, Data Devices, Electrical Fixtures, Communication Devices, Structural Foundations

Table 6
Definitions regarding the Precision–Recall formula.

	Related	Non-related
Retrieved	True Positive, TP	False Positive, FP
Not Retrieved	False Negatives, FN	True Negatives, TN

3.2.2. F-measure

While there's no necessary connection the precision and recall rate, in the actual retrieval process, these two indicators are mutually constrained. The F-measure metric is a combination of two indicators, denoted as:

$$F = \frac{\left(\beta^2 + 1\right) PR}{\beta^2 P + R}. \tag{10}$$

The parameters are determined according to specific needs and are usually set to 1. In general, the larger the value of F-measure, the better the retrieval performance.

3.2.3. Discounted cumulative gain

The DCG metric is also a widely used metric for retrieval accuracy which considers both the relevance and the location of the relevant results in the retrieved list. The formula can be denoted as:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i} \tag{11}$$

where DCG_p denotes the DCG value of the model located in the pth position. rel_i denotes the relevance of the query model to the ith model in the retrieved results list. We use the binary method to score the relevance of the retrieval results. When a model in the retrieval result is of the same type as the query model in the benchmark, the *rel* value of the model is set to 1 and vice versa. We use the calculated average DCG score to measure the performance of each search.

In general, the larger the value of DCG, the better the retrieval performance.

3.3. Comparison with Tversky

We compare our proposed similarity algorithm with the traditional Tversky similarity method. Experiments are based on the benchmark for the four resource libraries, Arcat, Autodesk Seek, BIMobject, and a library of all components from the three libraries (denoted as **All**). Each model in the repository is utilized as a query model. All component in the same category as the upload component in the benchmark are considered the positive set. That is, if the component in the search result is in the same category as the upload component in the benchmark, the retrieved result is correct. The importance of the attribute information is verified by the evaluation results of the three evaluation metrics described in Section 3.2.

3.3.1. Precision–Recall

Fig. 7 shows PR curve comparison using the proposed similarity measure and the Tversky similarity measure for the four libraries. It can be seen from the PR graph that the PR curve using the proposed similarity measure method is more convex than the PR curve of the Tversky similarity measure method alone. And for all the four libraries, the PR curve using the proposed measurement at the $(0.1, 1)$ coordinate is closer to 1, it can be seen that the proposed information-based similarity measure yields better performance.

3.3.2. F-measure

Table 7 shows the comparison results using the F-measure metric based on the four libraries. A larger F-measure value indicates a better retrieval performance. As can be seen from Table 7, our proposed similarity measurement has overall higher F-measure scores than the Tversky similarity alone.

Table 7
Comparison of F-measure between our proposed algorithm with Tversky similarity across all four BIM component libraries.

Method/recall	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Arcat(proposed)	0.18	0.32	0.44	0.54	0.61	0.67	0.71	0.74	0.73	0.21
Arcat(Tversky)	0.17	0.31	0.43	0.52	0.57	0.60	0.64	0.63	0.63	0.11
AutodeskSeek(proposed)	0.17	0.28	0.36	0.40	0.44	0.47	0.46	0.46	0.38	0.27
AutodeskSeek(Tversky)	0.16	0.27	0.34	0.39	0.44	0.44	0.43	0.43	0.37	0.20
BIMobject(proposed)	0.17	0.30	0.39	0.46	0.50	0.51	0.50	0.49	0.46	0.27
BIMobject(Tversky)	0.16	0.27	0.34	0.38	0.40	0.38	0.37	0.31	0.26	0.13
All(proposed)	0.17	0.30	0.38	0.44	0.48	0.50	0.51	0.49	0.46	0.18
All(Tversky)	0.16	0.28	0.36	0.41	0.44	0.45	0.45	0.42	0.36	0.09

Table 8
Comparison of F-measure between our proposed algorithm with geometric-based similarity across all four BIM component libraries.

Method/recall	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Arcat(proposed)	0.18	0.32	0.44	0.54	0.61	0.67	0.71	0.74	0.73	0.21
Arcat(geometric)	0.14	0.21	0.22	0.23	0.22	0.22	0.23	0.23	0.23	0.22
AutodeskSeek(proposed)	0.17	0.28	0.36	0.40	0.44	0.47	0.46	0.46	0.38	0.27
AutodeskSeek(geometric)	0.14	0.20	0.23	0.24	0.24	0.25	0.26	0.26	0.26	0.26
BIMobject(proposed)	0.17	0.30	0.39	0.46	0.50	0.51	0.50	0.49	0.46	0.27
BIMobject(geometric)	0.15	0.18	0.20	0.21	0.22	0.23	0.23	0.23	0.24	0.24
All(proposed)	0.17	0.30	0.38	0.44	0.48	0.50	0.51	0.49	0.46	0.18
All(geometric)	0.15	0.21	0.23	0.24	0.24	0.24	0.24	0.25	0.25	0.25

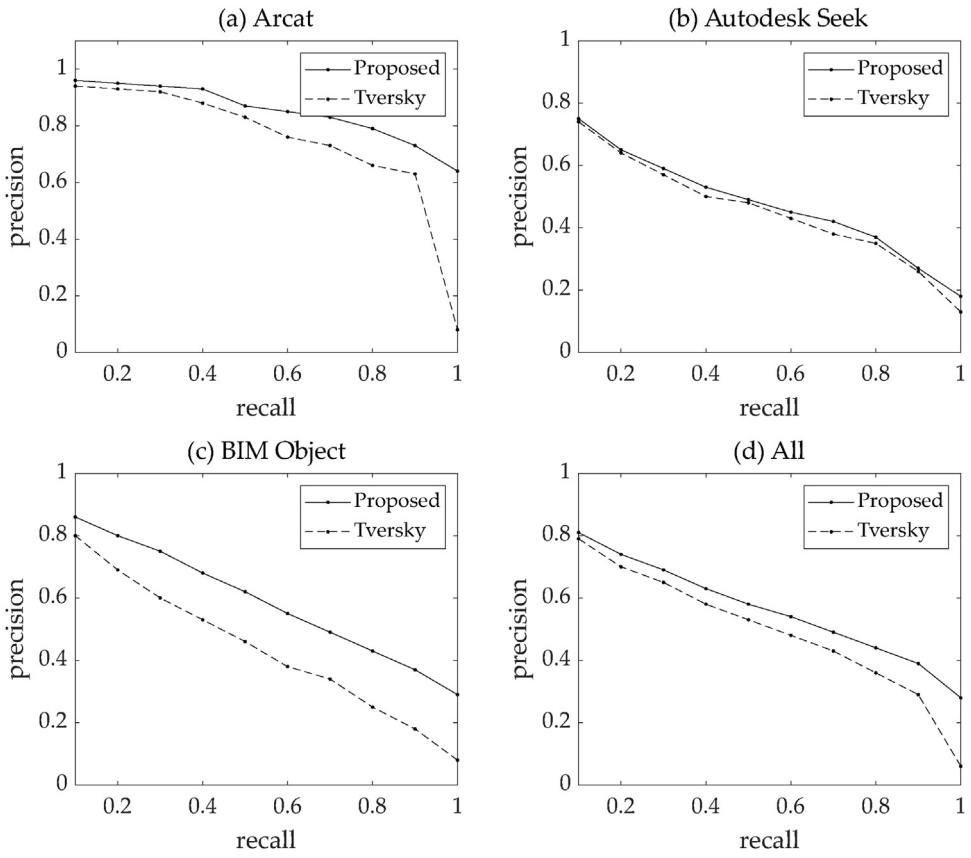

Fig. 7. Precision–Recall comparison between the proposed algorithm and Tversky on various BIM component libraries.

3.3.3. Discounted cumulative gain

For the four resource libraries, the DCG comparison results using the proposed method and the Tversky similarity measure alone are shown in Table 9. As can be seen from Table 9, regardless of which resource library, the value of DCG obtained by the proposed method is higher than the value of DCG obtained by using Tversky's similarity measure alone, therefore proving the better performance of our proposed method.

The experimental results show that the proposed method combining information content and Tversky similarity has superior retrieval effect. This is because the method incorporates the attribute information, and the effect of the information is to enlarge the effect of those attributes with higher discrimination, and to lower the effect of those attributes with lower discrimination.

3.4. Comparison with geometric based method

Our proposed similarity measure is also compared with the geometric similarity matching algorithm, which considers only the geometric information of the BIM model. In the experiment, for the four resource libraries of Arcat, Autodesk Seek, BIMobject and a library of all components from the 3 libraries (denoted as **All**). The three evaluation methods of PR curve, F-measure and DCG measure are used to evaluate the search results. Fig. 8 shows the comparison of the PR curves, 8 shows the comparison of the F-measures, and Table 9 shows the comparison of the DCG measures. By comparing the results, it can be seen that the BIM component attribute similarity measure based on information content and Tversky yields superior performance than the geometric matching method.

Table 9
Comparison of DCG_{avg} between our proposed algorithm, Tversky similarity and geometric-based similarity.

BIM library	Proposed	Tversky	Geometric
Arcat	**1.18**	0.27	0.48
Autodesk	**1.16**	0.22	0.23
BIMobject	**0.98**	0.14	0.30
All	**1.07**	0.12	0.32

4. Conclusion

In this work, we first propose a method for measuring the similarity of BIM component attributes based on information content and Tversky for two BIM components. This method combines the attribute information of the BIM component itself and combines it with the traditional Tversky similarity model in the attribute similarity measure. Then, based on the similarity measure method, a BIM component retrieval method based on model query is proposed. This method can be used when the user's needs are more complex and have old models or model sketches on hand. In addition, the retrieval results can be continuously optimized, that is, the user can query a similar model after retrieving a model, thereby narrowing the search range to find a suitable model more quickly. In order to verify the performance of the proposed similarity measure, we compare it with the traditional Tversky similarity measure and the geometric similarity matching method in the content-based attribute similarity measure. The four BIM libraries of Arcat, Autodesk Seek, BIMobject and a library consists of all components from the 3 libraries, and the three metrics of PR curve, F-measure metric and DCG metric are used to validate the performance. The experimental results show that our proposed method based on

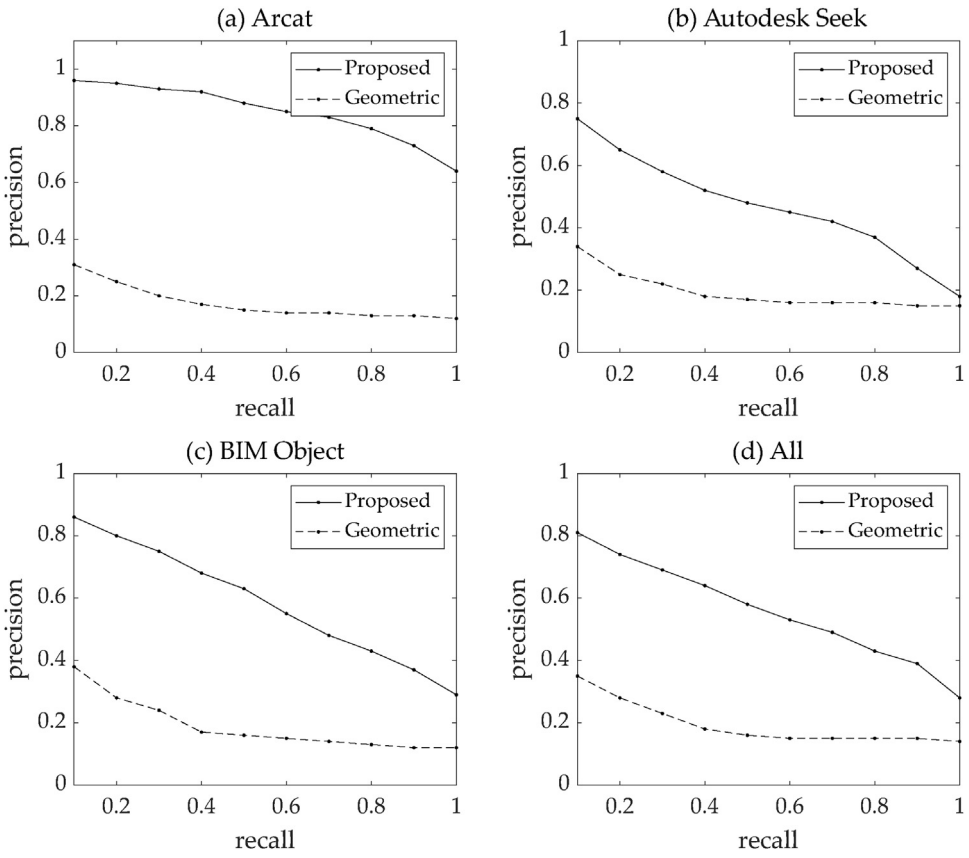

Fig. 8. Precision–Recall comparison between the proposed algorithm and geometric-based similarity on various BIM component libraries.

information and Tversky BIM component attribute similarity measure yields superior results.

Authors' contributions

Nanxing Li: methodology, software, investigation, visualization, writing-original draft, writing-review & editing.

Qian Li: conceptualization, methodology, software, investigation, writing-original draft.

Yu-Shen Liu: writing-review & editing, supervision.

Wenlong Lu: writing-review & editing.

Wanqi Wang: writing-review & editing.

Conflict of interest

None declared.

Acknowledgement

Project item: This work was supported by National Key R&D Program of China (2018YFB0505400). and the major R&D plan of China Railway Group (K2018G055).

References

Eastman, C., Teicholz, P., Sacks, R., Liston, K., 2011. BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2nd Edition. John Wiley and Sons, Hoboken, NJ.

Heaton, J., Parlikad, A.K., Schooling, J., 2019. Design and development of BIM models to support operations and maintenance. Comput. Ind. 111, 172–186.

Han, S.N., Lee, G.M., Crespi, N., 2014. Semantic context-aware service composition for building automation system. IEEE Trans. Ind. Informatics 10 (1), 752–761.

Kim, J.B., Jeong, W., Clayton, M.J., Haberl, J.S., Yan, W., 2015. Developing a physical BIM library for building thermal energy simulation. Autom. Constr. 50, 16–28.

Moreno, M.V., Terroso-Saenz, F., Gonzalez, A., Valdes-Vela, M., Skarmeta, A.F., Zamora-Izquierdo, M.A., Chang, V., 2017. Applicability of big data techniques to smart cities deployments. IEEE Trans. Ind. Informatics 13 (2), 800–809.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., Ramani, K., 2005. Three-dimensional shape searching: state-of-the-art review and future trends. Comput.-Aided Des. 37 (5), 509–530.

Han, Z., Lu, H., Liu, Z., Vong, C.-M., Liu, Y.-S., Zwicker, M., Han, J., Chen, C.P., 2019. 3D2SeqViews: aggregating sequential views for 3D gGlobal feature learning by CNN with hierarchical attention aggregation. IEEE Trans. Image Process. 28 (8), 3986–3999.

Han, Z., Shang, M., Liu, Z., Vong, C.-M., Liu, Y.-S., Han, J., Zwicker, M., Chen, C.P., 2019. SeqViews2SeqLabels: learning 3D global features via aggregating sequential views by RNN with attention. IEEE Trans. Image Process. 28 (2), 658–672.

Wu, S., Shen, Q., Deng, Y., Cheng, J., 2019. Natural-language-based intelligent retrieval engine for BIM object database. Comput. Ind. 108, 73–88.

Lin, J.-R., Hu, Z.-Z., Zhang, J.-P., Yu, F.-Q., 2016. A natural-language-based approach to intelligent data retrieval and representation for cloud BIM. Comput.-Aided Civil Infrastruct. Eng. 31 (1), 18–33.

Fu, X., 2006. From 2D to 3D: information driven architectural design. World Architect. 9 (1).

Lu, H.-L., Wu, J.-X., Liu, Y.-S., Wang, W.-Q., 2019. Dynamically loading IFC models on a web browser based on spatial semantic partitioning. Visual Comput. Ind. Biomed. Art 2 (1), 1.

Gunn, T.G., 1982. The mechanization of design and manufacturing. Sci. Am. 247 (3), 114–131.

Ullman, D.G., 2004. The mechanical design process, 1997, McGraw-Hill, New York, NY, USA) Darlington, MJ and Cully SJ A model of factors influencing the design requirement. Des. Stud. 25 (4), 329–350.

Leizerowicz, W., Bilgic, T., Lin, J., Fox, M.S., 1996. Collaborative design using WWW. Proceedings of the WET-ICE'96.

Tversky, A., 1977. Features of similarity. Psychol. Rev. 84 (4), 327.

Resnik, P., 1995. Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), 448–453.

Gao, G., Liu, Y.-S., Wang, M., Gu, M., Yong, J.-H., 2015. A query expansion method for retrieving online BIM resources based on industry foundation classes. Autom. Constr. 56, 14–25.

Gao, G., Liu, Y.-S., Lin, P., Wang, M., Gu, M., Yong, J.-H., 2017. BIMTag: concept-based automatic semantic annotation of online BIM product resources. Adv. Eng. Informatics 31, 48–61.

Zuccon, G., Koopman, B., Nguyen, A., Vickers, D., Butt, L., 2012. Exploiting medical hierarchies for concept-based information retrieval. Proceedings of the Seventeenth Australasian Document Computing Symposium, 111–114.
Salton, G., McGill, M.J., 1986. Introduction to Modern Information Retrieval. McGraw-Hill, Inc., NY, USA.
Akgül, C.B., Sankur, B., Yemez, Y., Schmitt, F., 2009. 3D model retrieval using probability density-based shape descriptors. IEEE Trans. Pattern Anal. Mach. Intell. 31 (6), 1117–1133.
Chen, Y.-L., Chiu, Y.-T., 2011. An IPC-based vector space model for patent retrieval. Inform. Process. Manag. 47 (3), 309–322.
Erk, K., 2012. Vector space models of word meaning and phrase meaning: a survey. Lang. Linguist. Compass 6 (10), 635–653.
Han, Z., Liu, Z., Vong, C.-M., Liu, Y.-S., Bu, S., Han, J., Chen, C.P., 2018. Deep spatiality: unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax. IEEE Trans. Image Process. 27 (6), 3049–3063.
Han, Z., Liu, Z., Vong, C.-M., Liu, Y.-S., Bu, S., Han, J., Chen, C.P., 2017. BoSCC: bag of spatial context correlations for spatially enhanced 3D shape representation. IEEE Trans. Image Process. 26 (8), 3707–3720.
Liu, Y.-S., Ramani, K., Liu, M., 2011. Computing the inner distances of volumetric models for articulated shape description with a visibility graph. IEEE Trans. Pattern Anal. Mach. Intell. 23 (12), 2538–2544.
He, T., Zhang, J., Lin, J., Li, Y., 2018. Multiaspect similarity evaluation of BIM-based standard dwelling units for residential design. J. Comput. Civil Eng. 32 (5), 04018032.
Pu, Y.-C., Du, W.-C., Huang, C.-H., Lai, C.-K., 2012. Invariant feature extraction for 3D model retrieval: an adaptive approach using Euclidean and topological metrics. Comput. Math. Appl. 64 (5), 1217–1225.
Han, Z., Liu, X., Liu, Y.-S., Zwicker, M., 2019. Parts4Feature: learning 3D global features from generally semantic parts in multiple views. International Joint Conference on Artificial Intelligence (IJCAI).
Han, Z., Wang, X., Vong, C.-M., Liu, Y.-S., Zwicker, M., Chen, C., 2019. 3DViewGraph: learning global features for 3D shapes from a graph of unordered views with attention. International Joint Conference on Artificial Intelligence (IJCAI).
Han, Z., Shang, M., Liu, Y.-S., Zwicker, M., 2019. View inter-prediction GAN: unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions. AAAI Conference on Artificial Intelligence.
Han, Z., Wang, X., Liu, Y.-S., Zwicker, M., 2019. Multi-Angle Point cloud-VAE: unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. IEEE International Conference on Computer Vision (ICCV).
Zhou, X., Qiu, Y., Hua, G., Wang, H., Ruan, X., 2007. A feasible approach to the integration of CAD and CAPP. Comput.-Aided Des. 39 (4), 324–338.
Herrmann, J.W., Singh, G., 1997. Design similarity measures for process planning and design evaluation. MARYLAND UNIV COLLEGE PARK DEPT OF MECHANICAL ENGINEERING, Tech. Rep.
Pan, X., Zhang, S., Ye, X., 2009. Advance in 3D model semantic retrieval. Chin. J. Comput. (6), 3.
Zhou, Y.-W., Hu, Z.-Z., Lin, J.-R., Zhang, J.-P., 2019. A review on 3d spatial data analytics for building information models. Arch. Comput. Methods Eng., 1–15.
Kiryakov, A., Popov, B., Terziev, I., Manov, D., Ognyanoff, D., 2004. Semantic annotation, indexing, and retrieval. Web Semant.: Sci. Serv. Agents World Wide Web 2 (1), 49–79.
Berners-Lee, T., Hendler, J., Lassila, O., et al., 2001. The semantic web. Sci. Am. 284 (5), 28–37.
Liu, H., Liu, Y.-S., Pauwels, P., Guo, H., Gu, M., 2017. Enhanced explicit semantic analysis for product model retrieval in construction industry. IEEE Trans. Ind. Informatics 13 (6), 3361–3369.
BuildingSMART, 2014. Industry Foundation Classes (IFC), Available from: http://www.buildingsmart-tech.org/specifications/ifc-overview/.
Lin, Y.-H., Liu, Y.-S., Gao, G., Han, X.-G., Lai, C.-Y., Gu, M., 2013. The IFC-based path planning for 3D indoor spaces. Adv. Eng. Informatics 27 (2), 189–205.
Sun, J., Liu, Y.-S., Gao, G., Han, X.-G., 2015. IFCCompressor: a content-based compression algorithm for optimizing Industry Foundation Classes files. Autom. Constr. 50, 1–15.
Shi, X., Liu, Y.-S., Gao, G., Gu, M., Li, H., 2018. IFCdiff: a content-based automatic comparison approach for IFC files. Autom. Constr. 86, 53–68.
Staub-French, S., Nepal, M.P., 2007. Reasoning about component similarity in building product models from the construction perspective. Autom. Constr. 17 (1), 11–21.
Miller, G.A., 1995. WordNet: a lexical database for English. Commun. ACM 38 (11), 39–41.