Enhancing performance-based generative architectural design with sketch-based image retrieval: a pilot study on designing building facade fenestrations

Article *in* The Visual Computer · August 2022

DOI: 10.1007/s00371-021-02170-x

Shenghuan Zhao
Suzhou University of Science and Technology

Luo Wang
Beijing University of Technology

The Visual Computer
https://doi.org/10.1007/s00371-021-02170-x

ORIGINAL ARTICLE

Enhancing performance-based generative architectural design with sketch-based image retrieval: a pilot study on designing building facade fenestrations

Shenghuan Zhao[1,2] · Luo Wang[3] · Xueming Qian[3] · Jianping Chen[1,2,4]

Accepted: 22 May 2021

Abstract

By coupling parametric modelling, building performance (like energy efficiency) simulation, and algorithmic optimization, performance-based generative architectural design (PGAD) can automatically generate lots of high-performance architectural design solutions. Although it is 'performance-based', the final selection of a real design project still needs to consider the aesthetics of design choices. However, due to the overwhelming number of design choices generated by PGAD, it is difficult for designers to choose the most favourable one from them. Therefore, the current study tries to integrate the technology of sketch-based image retrieval (SBIR) into the selecting stage of PGAD. Rather than navigating alternatives one from another and getting lost, designers can directly find the most aesthetically preferred one by inputting his/her hand-drawn design. A design project of fenestrating a multiple-floor office building is used to demonstrate this method and test three SBIR algorithms: Angular radial partitioning (ARP), Angular radial orientation partitioning (AROP), and Sketch-A-Net model (SAN). Test results show that AROP performs the best among these three algorithms. Its retrievals are most similar to inquiry images drawn by architects. Meanwhile, performances of AROP with different template combinations are also rated. After that, AROP with the best template is also tested with incompletely drawn inquiry images. In the end, investigation results are validated by another building façade design case. The current study automates the PGAD process stepwise, making it more applicable to real design projects.

Keywords Computer-aided design · Geometric computation · Design method · Artificial intelligence · Sketch-based image retrieval (SBIR)

1 Introduction

With algorithmic optimization automatically searching a design space and generating design alternatives, generative techniques have been increasingly used in the architectural design field. Architects and researchers use this method to obtain design alternatives, such as building shapes and floor plans. A stepwise, performance-based generative architectural design (PGAD) is a design method coupling parametric modelling, building performance simulation, and algorithmic optimization to automatically output tens and thousands of design solutions, some of which have high building performances, aspects like energy-efficient, financial cost-optimal, etc. [1–3]. For designing high-performance buildings, with the help of algorithmic optimization, the PGAD method is much more efficient than the traditional design approach, which requires manual iteration but only exploring a very limited design space.

Shenghuan Zhao and Luo Wang equally contributed to the paper and considered as co-first authors.

✉ Jianping Chen
alanjpchen@yahoo.com

1 School of Architecture and Urban Planning, Suzhou University of Science and Technology, Suzhou, China

2 Jiangsu Key Laboratory of Intelligent Building Energy Efficiency, Suzhou University of Science and Technology, Suzhou, China

3 School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an, China

4 School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou, China

Published online: 11 June 2021

Although PGAD is much faster and more effective in generating design solutions than human brains, selecting design solutions in its post-processing stage is still time and effort consuming. The reason is that aesthetics and human subjective preferences also play a more important role during the decision-making process. While there have been already developed visualization gadgets like DesignExplorer [4] or Project Refinery [5], designers can check the physical appearances of some design solutions by setting a specific domain on building performances, a kind of filtering. However, they still have to spend a lot of time navigating and selecting among many design solutions.

To speed up the navigating and selecting process, some scholars used clustering algorithms, one pattern of unsupervised learning which organizes un-labelled data into similarity groups, to cluster design solution alternatives [6–12]. By clustering design solutions before looking into details of singular solutions, designers can navigate alternatives from group to group to get an overall picture of all the design solutions. Commercial optimization software like modeFRONTIER and other open-source optimizers like Wallacei [13] have already offered this clustering function.

However, these studies and tools still have limitations. Firstly, except for the work done by Rodrigues et al. [8] which clustered solutions' phenotypes, all the others clustered genotypes of design solutions or values of building performance objectives [10, 14]. In other words, most of the studies in the literature only clustered parameters' values of generating design choices or simulation results of building performances, rather than directly clustering design choices' physical appearances. However, geometric variables in the generating stage are unable to fully describe what people really can perceive when seeing buildings in the real world, let alone values of building performance which does not identify the physical appearance and aesthetics at all. In conclusion, current methods did not offer people direct perceptions to design solutions' physical appearances, which requires critical attention in the design process.

Secondly, nearly all previous studies only clustered optimal design solutions or not optimal solutions but have been explored by the optimization algorithm. However, the algorithms always do not explore the whole design space, which contains all possible design solutions. It means that many sub-optimal or non-explored design solutions but with more preferred appearances are ignored. Although performance plays a crucial role in PGAD, aesthetic considerations actually are still necessary for real architectural design projects. However, it is impossible to cluster the whole design space using their methods, because geometric parameters are evenly sampled, and values from these variables do not have any variation for clustering.

Thirdly, architects' subjective preferences on aesthetics, which are crucial for them to complete a project, were not adequately embodied in the clustering process. The workflow in these studies became that they offered several groups of design choices, and architects checked and selected one from them. It somehow reduces architects' initiatives because it is different from the traditional way that architects proactively propose several choices and compare them. In conclusion, (image) clustering can aid the selection phase of PGAD to a certain extent, but it is still difficult for architects to make decisions among tens and thousands of design solutions.

Similar to image clustering, sketch-based image retrieval (SBIR) is another research topic in the field of computer vision. SBIR allows people to use simple sketches as queries to find similar images from storage. Meanwhile, fast hand drawing is the most convenient method for architects to express and record their design ideas. Therefore, the current study tries to couple PGAD and SBIR together, empowering architects to directly find the design choice with aesthetically wanted appearance. This study attempts to contribute the two fields from four aspects:

- turning the navigating stage of PGAD into an intention-based selection,
- turning the machine-dominated digital design workflow into a machine-human interactive process,
- saving the unnecessary time expenditure for architects when using PGAD,
- investigating detailed parameter-tuning of SBIR algorithms.

On top of the current introductory section, this paper is divided into the other five main parts. SBIR itself and its integration with the design field are roughly reviewed in Sect. 2. After that, the research methodology of this study is introduced in Sect. 3, including six sub-sections: the overall workflow, gallery dataset generation, inquiry dataset generation, implementing SBIR with complete inquiry images, implementing SBIR with incomplete inquiry images, and evaluation results. Thenceforth, Sect. 4 illustrates results and analyses them from several perspectives. Section 5 introduces Building Case 2, used to validate investigation findings obtained from the previous case. In the end, Sect. 6 draws up the whole study and sheds light on future research possibilities.

2 Background

2.1 A literature on SBIR

Searching among relevant images according to a user's inquiry, image retrieval is a popular research topic in the field of image processing. Apart from the widely used text-based image retrieval [15–18] and content-based image retrieval [19–22], sketch-based image retrieval (SBIR) [23–34] also benefits the users' image search, by using computer vision algorithms to help people find the most aesthetically wanted images.

SBIR algorithms can be roughly classified into two categories: SBIR algorithms based on geometric features and SBIR algorithms based on deep learning features. There are plenty of SBIR algorithms based on geometric features [23–28]. Generally speaking, they are implemented by two steps: edge detection and feature matching [35]. Edge detection transforms natural (real) images that contain abundant semantic information into edge maps that only have simple lines and white background. Edge detection is used to bridge the image domain gap between sketches and natural (real) images. Then, a common geometric feature is set for both sketches and edge maps, and their similarity can be calculated via the common feature.

The other type of SBIR method uses a deep learning model to simultaneously learn the semantic information of both sketches and natural images [29–34, 36]. Deep learning techniques, especially convolutional neural networks (CNN), frequently perform better than traditional machine learning techniques. When using deep learning, sketches and natural images are directly treated as the input during feature extraction, without the need of an edge map as a bridge.

For example, Siamese network and Triplet network are two CNN-based deep learning techniques that are commonly used to do sketch recognition and SBIR. The former network trains two CNNs together [30, 33], one for training sketches and another for training natural images. Triplet network [30, 32] is a network with three CNNs. Sketch, natural images relevant to this sketch, and natural images irrelevant to this sketch are inputs of these three CNNs. Besides these two CNNs, there are also CNN-based methods for sketch recognition. Yu et al. [37] proposed a CNN named Sketch-A-Net, which was explicitly designed for sketch recognition and outperformed human brains in those authors' experiments.

2.2 A review on integrating SBIR with design processes

What has been reviewed is solely from the SBIR side. In the current sub-section, human intervention and specific SBIR applications in the design field are going to be surveyed. Twenty years ago, there already have been discussions on bringing qualitative and quantitative design criteria together [38]. In Takagi's survey, he introduced how humanized evolutionary optimization is applied to various kinds of scenarios. Based on Takagi's theory, Brintrup et al. [39] developed an optimization framework that couples human preference of the evolutionary design process. This framework was tested with an ergonomic chair design experiment. Similarly, for enhancing evolutionary design in architecture, aesthetic-based fitness measure combing quantitative and qualitative criteria was applied to explore a set of 3D-shapes solutions [40]. In addition, researchers used quantitative ways to promote the creativity of a generative design system [41].

Different from those studies which tried to insert subjective intervene in the middle of design processes, SBIR allows designers to interact with a larger number of alternatives at the end of design processes. For applications specifically related to architecture, SBIR was mainly implemented to retrieve architectural floor plans from a repository [42–46]. They developed a platform named a.SCatch system. Designers can search for wanted building layouts from a repository by sketching a schematic abstraction of a floor plan. A deep multilayer convolutional network was used to retrieve architectural images [47]. Another category of related studies utilized SBIR to retrieve 3D design geometries. For example, Wessel et al. [48] used embeddings of attributed subgraphs to retrieve 3D building models. In his thesis, by using a supervised learning approach, a new meta-descriptor was developed to retrieve 3D models [49]. Some researchers also discussed selecting views when using sketches to retrieve 3D models [50].

However, except for the DreamSketch, which is a novel 3D design interface enabling designers to explore a range of functional 3D designs by sketching design intentions [51], there seems still no other similar study directly taking the SBIR into the PGAD field. Moreover, designers' intents are always fuzzy at the beginning of a design project. Therefore, the main idea of the current study is to initially apply SBIR into post-PGAD. This integration would be mutually useful and beneficial to each side. Based on the first author's previous study [1], building façade design is the main subdomain in the field of PGAD. Therefore, as the pilot study communicating PGAD and SBIR, the present research also focuses on the generative design of building façade fenestrations. The foremost challenge herein is that traditional SBIR research was conducted with gallery datasets composed of different images, easier for algorithms to find the wanted image. However, PGAD generates a gallery dataset with quite similar images.

3 Methodology

3.1 The overall workflow of SBIR research

As shown in Fig. 1, the SBIR's whole workflow is composed of four main steps:

- Generate the gallery dataset.
- Generate the inquiry dataset.

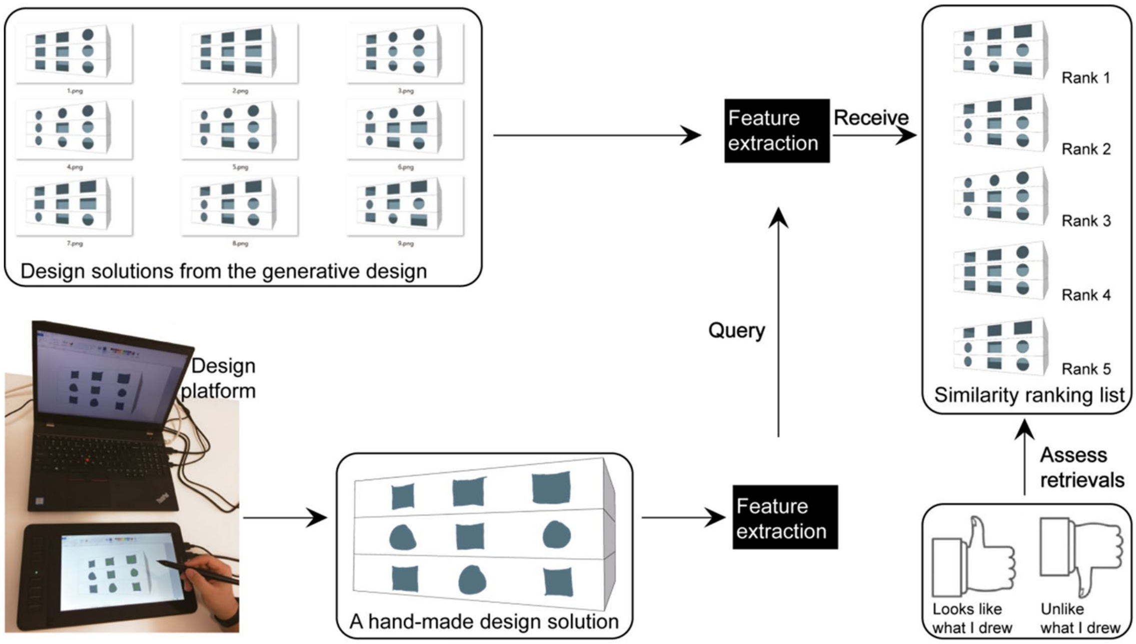


Fig. 1 The general workflow of this SBIR study

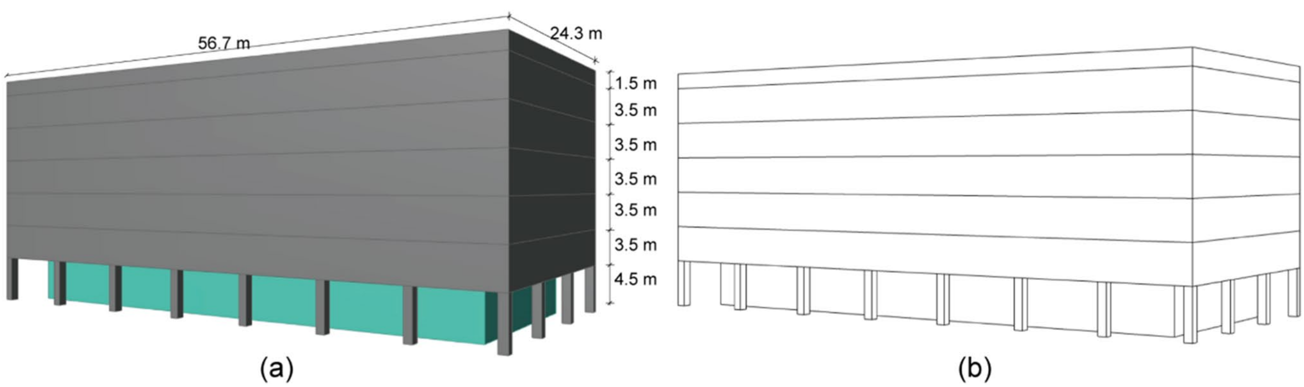


Fig. 2 The building size (**a**); the line drawing prepared for designers (**b**)

- Apply SBIR algorithms to extract image features in two datasets, calculate image similarities, and visualize searching results.
- Evaluate SBIR results and assess SBIR algorithms' performances.

3.2 Gallery dataset generation

SBIR algorithms are applied to the case of designing one office building's fenestration configurations. A representative six-floor office building (Fig. 2a), with no fenestration on facades, is taken as this case study. Its ground floor is a restaurant enclosed with full glass curtains, and facades of the top five floors are going to be fenestrated. Window wall ratios on four orientations are 0.28 for the north, 0.20 for the west, 0.42 for the south, and 0.36 for the east. These data are obtained by an algorithmic optimization that tried to balance the energy consumption and indoor visual comfort. More related information can be found in another publication [52]. A parameterization system is applied to blank facades for generating detailed fenestration solutions, trying to balance

Table 1 The parameterization system of Case 1

Variables	Ranges	Stepwise for PGAD	Stepwise herein
Number of windows	{1, 2, 4}	–	–
Window height (m)	[1.6, 2.4]	0.01	0.05
Sill height (m)	[0.15, 1.05]	0.05	0.1

visual comfort and the construction material cost. Design alternatives are generated by three variables (Table 1): the number of windows between two columns, window height, and sill height.

Theoretically, as the current study is aimed to serve the PGAD, the gallery dataset should be inherited from the real PGAD approach. However, building performance simulation and algorithmic optimization will not be conducted herein, and the design space will be a little different from PGAD, due to the following three reasons:

- Firstly, PGAD pursues design solutions on the Pareto surface (or frontier) which identify relatively optimal performances. However, non-dominated design solutions and non-explored ones from the design space also should be considered in the selection phase, because they may be aesthetically preferred.
- Secondly, due to the small variation of window height that may be imperceivable by human eyes from the whole perspective of the building, the stepwise of parameters herein actually is larger than set in the same case done in another paper, which was more focused on the building performance simulation and optimization [52].
- Thirdly, they are not the specific focal point herein. Readers can find this PGAD process in the previous publication [52]. Simultaneously, the current study still can be applied to real PGAD processes, which have stages of simulation and optimization.

The 3D modelling software Rhino, with its extension Grasshopper, and two plugins of Grasshopper are used here to generate the gallery dataset. The code to implement this process is illustrated in Fig. 3a. Grasshopper offers a visual programming platform, allowing users to code by connecting coded components, without knowing mainstream programming languages. As one Grasshopper plugin, Honeybee [53] is used for parametrically creating windows and taking snapshots of the screen of software Rhino. Every time when parameter values are changed and a new design choice is created, Honeybee takes one screenshot as a gallery image. Two individual gallery cases are illustrated in Fig. 3b,c. Another Grasshopper plugin Colibri [54] automatically iterates the parametric

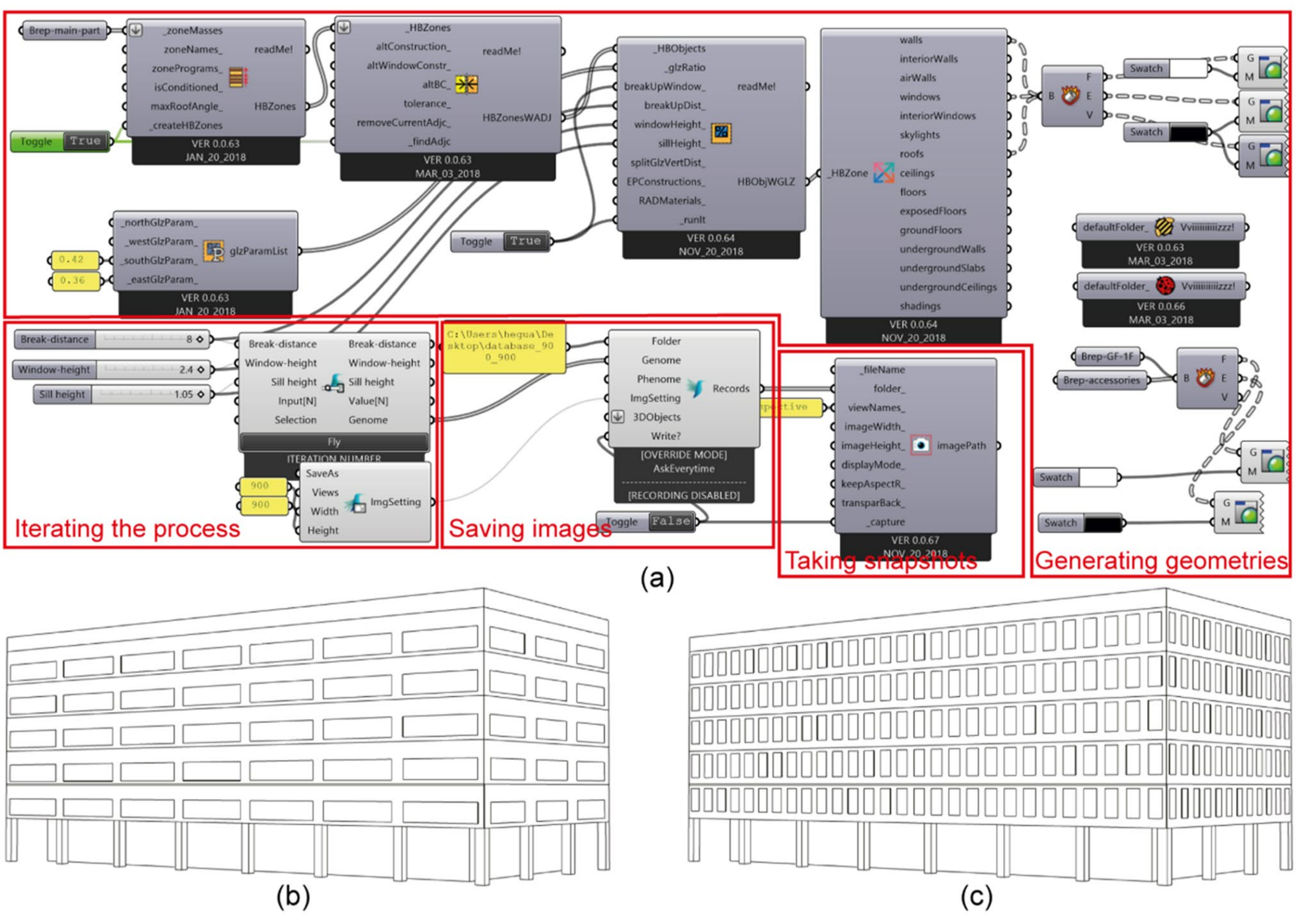


Fig. 3 The Grasshopper code of gallery dataset generation (**a**); two individual cases from the gallery dataset (**b, c**)

modelling process and saves all screenshots during the whole process. Due to the need for SBIR algorithms which will be used in the retrieving step, the image resolution is set at 900 pixels × 900 pixels. In the end, after 510 design alternatives are all automatically saved as JPG files in a folder, the gallery image dataset is prepared for the SBIR research.

3.3 Inquiry image dataset generation

On the other side, inquiry pictures are drawn by designers using a liquid–crystal display (LCD) screen and an LCD pen. The LCD screen used herein is from the manufacturer Gaomon (type GM 116 HD). However, designers also can use an iPad and an iPencil instead. By connecting it to a computer, the LCD screen allows users to monitor the computer screen and directly operate any graphics processing software on it. Although many graphics processing software is adaptive to this device, authors adopt the Autodesk Sketchbook, version 8.6.0. This software can mimic the degree of ink bleeding caused by pushing the pen with different strengths. Therefore, users can feel like drawing with a real pen on a real piece of paper. This feeling helps architects accept digital design methods.

Thirty students from architecture major are invited to create the inquiry dataset and assess algorithms' performances. Each participant is provided with a picture of the blank facade building (Fig. 2b), which is in the same visual perspective as gallery images. Students are required to design two different solutions of fenestration geometries, drawing with electronic tools introduced above (Fig. 4a). Windows should be rectangles or squares and evenly positioned on facades, but designers still have the freedom of deciding the number of windows existing between two columns, the window height, and the sill height. In the end, sixty manually drawn images of facade fenestration (Fig. 4b–d) are stored in another folder as inquiry images.

3.4 Sketch-based image retrieval

The whole SBIR process includes two main steps: edge detection and edge similarity measurement. Edge detection finds and records both inquiry images and gallery images' edge pixels and their orientation information, while edge similarity measurement evaluates the similarity between those edge pixels. At first, two SBIR algorithms based on geometric features are used as retrieval methods. In addition, a deep-learning-based CNN, which has an excellent semantic understanding ability for sketches, is used as the third SBIR algorithm in the present research.

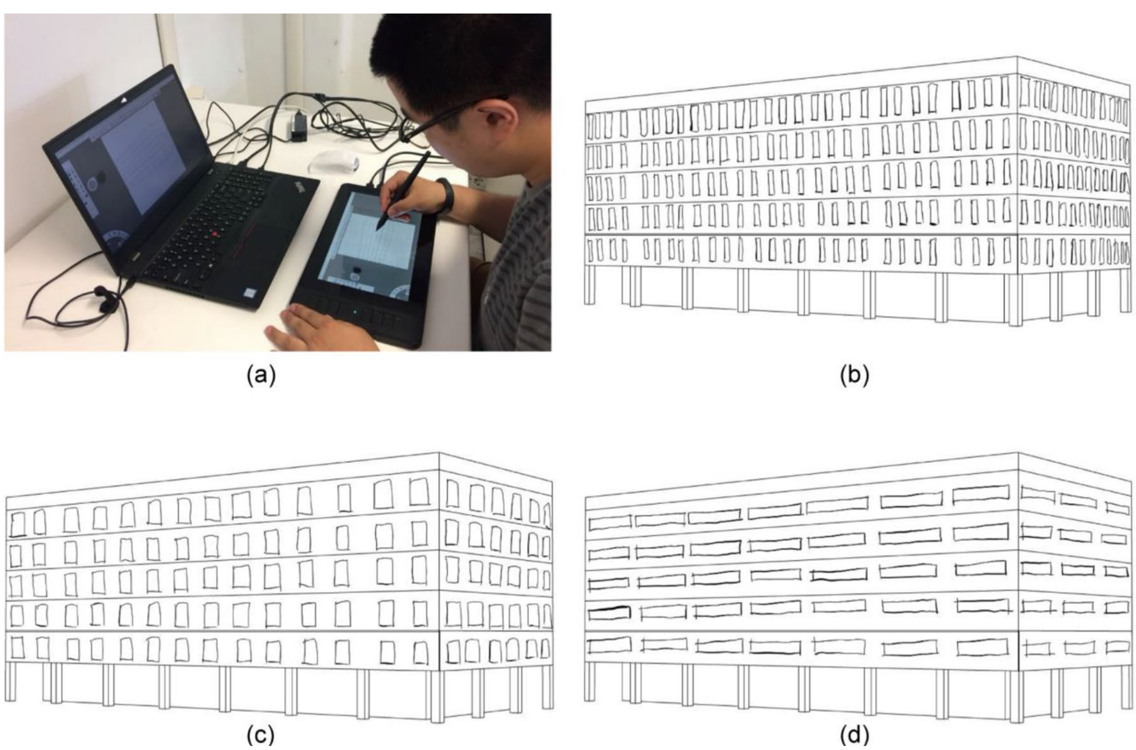


Fig. 4 A designer is creating inquiry images (**a**); three individual cases from the inquiry dataset (**b–d**)

3.4.1 Edge detection

Representing the main contours and boundaries of images, edge maps are of great importance in various image processing tasks. To get edge maps, edge detectors need to record the location and orientation of the edge pixels of both inquiry and gallery images. Canny detector [55] and Berkeley detector [56] are two commonly used edge extraction detectors to get edge pixels. However, when using these two detectors to extract edge maps of current inquiry images and gallery images, edge maps' visual quality was unsatisfying. The reason is that, for gallery images (e.g. Figure 3b) and inquiry images (e.g. Figure 4b), rectangles inside them are very close to each other. As a result, these two detectors are unable to identify most black pixels of both inquiry and gallery images as edge pixels, resulting in lousy retrieval performance.

Therefore, to accomplish the retrieval task, a new edge detector is specifically developed with two steps: edge pixel detection and edge orientation setting. For the first step, since inquiry and gallery images herein are visually similar to edge maps, sketch-like images comprising of black lines and white background, all black pixels of these images are directly regarded as edge pixels. That is to say, in generated edge maps of both gallery and inquiry images, edge pixels are black pixels of those images, and the background pixels are white pixels of those images. The second step is the edge orientation setting, setting an orientation for each edge pixel. Inquiry images and gallery images herein primarily contain two directions: horizontal and vertical. So, three types of edge pixel orientations are set: horizontal, vertical, and others.

When human intuitively observe and detect several consecutive horizontal or vertical pixels, these pixels are regarded as a horizontal line or a vertical line. Accordingly, we set a vertical detection template that contains three consecutive horizontal pixels and a horizontal detection template that contains five consecutive vertical pixels. Orientations of all edge pixels are set based on these two templates. Firstly, the vertical template moves throughout an edge map. During its moving, if pixels from the edge map are overlapped by the vertical template, these edge pixels' orientation is set to 'vertical'. Afterwards, the horizontal template experiences similar operations, and some edge pixels' orientation becomes 'horizontal'. Finally, the orientation of the left edge pixels, neither 'vertical' nor 'horizontal' ones, is marked as 'others'. Figure 5 is an example of edge detection steps.

3.4.2 SBIR algorithms measuring similarities

To perform the SBIR task, three different SBIR algorithms are tested and compared: Angular radial partitioning (ARP) [57], Angular radial orientation partitioning (AROP) [24] and the Sketch-A-Net model (SAN) [58]. ARP [57] and AROP are based on geometric features, while the Sketch-A-Net is a CNN (Convolutional Neural Network) Caffe model through deep learning.

ARP divides an edge map into $M \times N$ sectors and records the number of edge pixels in each sector. M is the number of radial partitions, and N is the number of angular partitions [57]. As a result, each sector has an edge pixel number, and an $M \times N$ matrix can represent the feature matrix of the edge map. Based on the authors' previous study [24], the M is set

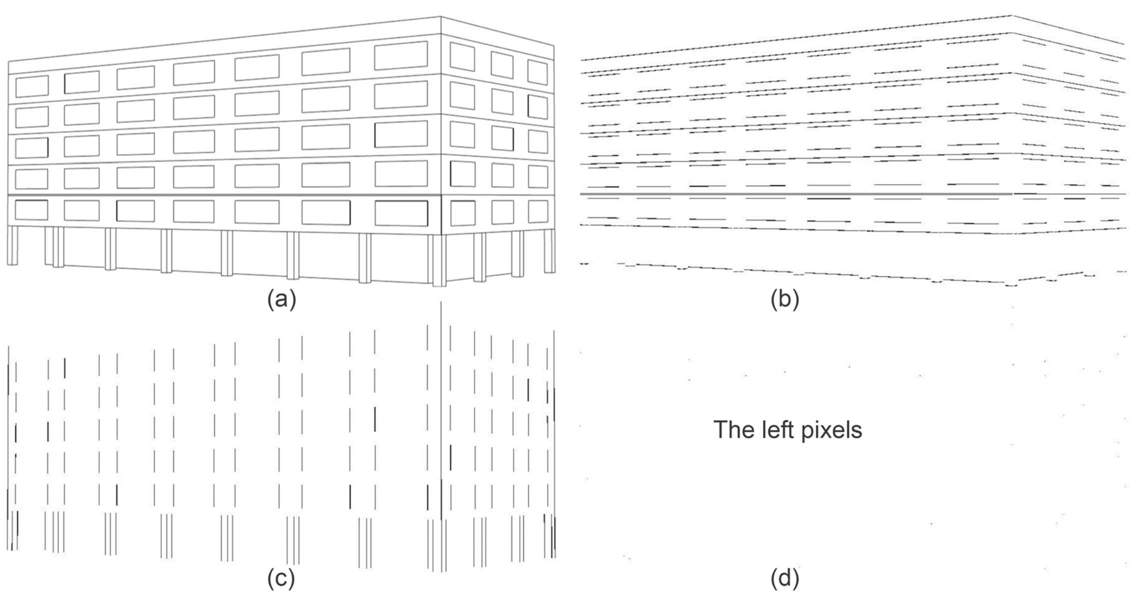


Fig. 5 The edge map (**a**); horizontal edge pixels (**b**); vertical edge pixels (**c**); other edge pixels (**d**)

to 8 and N is set to 4 herein. Each value in the matrix is the edge pixel number inside that specific sector. Finally, two edge maps' Euclidean distance is calculated by comparing the difference between the feature matrices of two images.

ARP merely takes the number of edge pixels into account, neglecting edge pixels' orientation. As an improved version of ARP, AROP not only divides an edge map into $M \times N$ sectors and considers the edge pixel numbers, but also recognizes orientations of those edge pixels [24]. As edge detection has three types of orientations for edge pixels, each sector has three edge pixel numbers: the number of horizontal edge pixels, the number of vertical edge pixels, and the number of the other edge pixels. Thus, the 'shape' of edge maps' feature matrix is $M \times N \times 3$. The values of this matrix represent the edge pixel number and orientation in one sector. For the first-round experiment, AROP has the same M & N settings as in ARP. Finally, the Euclidean distance between two feature maps is calculated, to identify the similarity between two edge maps.

Moreover, regarding pixels' orientations on an image, the authors investigate different template combinations for AROP. Both the horizontal template and the vertical template are designed into four types: 2 pixels, 3 pixels, 5 pixels, 8 pixels. One combination of a horizontal template and a vertical template is named as '*havb*', which means that the horizontal template in this combination has '*a*' pixels while the vertical template has '*b*' pixels. In total, there are 16 templates to assign orientations to edge pixels. Besides more combinations of h and v, the authors also conduct more experiments on values of M and N for AROP, trying to explore influences of different M and N values. M is set to 4, 8, 16, 32 respectively, and N is set to 2, 4, 8, 16 respectively.

At last, SAN [58], a deep-learning-based method specifically designed for sketches and performed greatly in the task of sketch recognition on the ImageNet database [59], is tested for the current experiment. We use this model as the third method to compare with the other two algorithms, because it is generally assumed that deep CNNs have a better ability of image semantic understanding than geometric features. SAN is composed of several convolutional layers, pooling layers, fully connected layers, and a SoftMax layer (Fig. 6). This trained CNN model is taken as the feature extraction tool. Instead of using edge maps like ARP and AROP, the input of this method is the original inquiry and gallery images. Those images are fed into the CNN model to get their deep features, where the deep features being extracted are the output of the 'fc7' layer. The image similarity is measured by calculating the Euclidean Distance between the deep features of two images.

3.5 Tests with incomplete inquiry images

The whole SBIR process introduced above is still time-consuming because designers need to completely draw all window drawings on facades. It would be better if the searching process can be initiated when just a part of the windows is drawn. Therefore, the authors try to improve workflow efficiency and save architects' time stepwise.

The main idea is to check the algorithm's effectiveness when replacing completely drawn inquiry images with incomplete images. As shown in Fig. 7a and Table 2, windows on facades are divided into six independent sections. After that, seven different scenarios are created to obtain incomplete inquiry images. For scenario 1, authors only use contents in section 1 and leave other sections blank to search for similar images. The other five scenarios are combinations of different sections, as listed in Table 2. For scenario 7, the authors randomly delete a part of windows on each inquiry image (Fig. 7b). As a result, deriving from one original design case, seven new incomplete inquiry images are generated, and there are 420 new inquiry images in total. The algorithm which has the best performances in

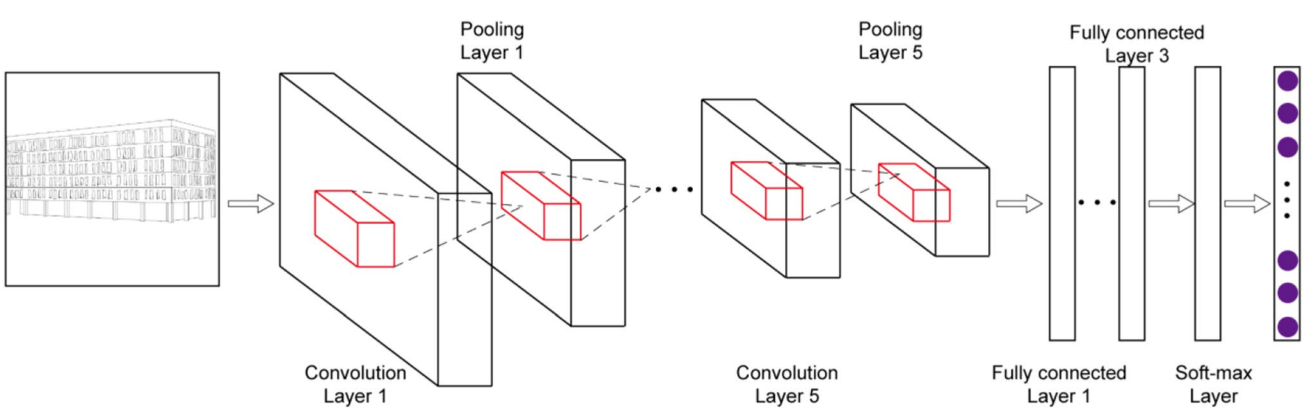


Fig. 6 The structure of SAN

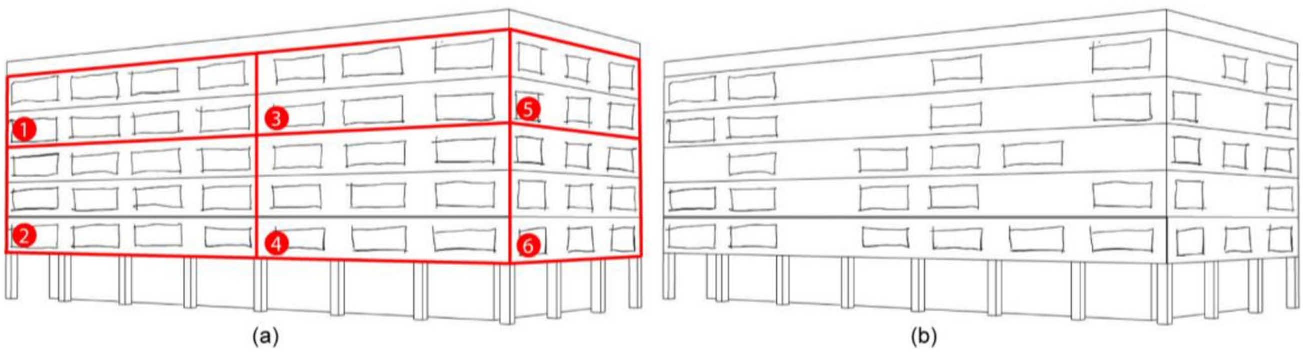


Fig. 7 Dividing the whole fenestration area into six individual sections (**a**); One of images which have randomly drawn windows (**b**)

Table 2 Six different scenarios to obtain incomplete inquiry images

Scenarios	Sections included in this scenario
Scenario 1	Section 1
Scenario 2	Section 1 + Section 2
Scenario 3	Section 1 + Section 2 + Section 3
Scenario 4	Section 1 + Section 2 + Section 3 + Section 4
Scenario 5	Section 1 + Section 2 + Section 3 + Section 4 + Section 5
Scenario 6	Section 1 + Section 3 + Section 5
Scenario 7	Images with randomly drawn windows

the previous test will be continually utilized for incomplete inquiry images.

3.6 Human-based evaluation and validation

After retrievals obtained from those experiments, they are assessed by designers who contribute to the inquiry image dataset. This assessing process is divided into four phases:

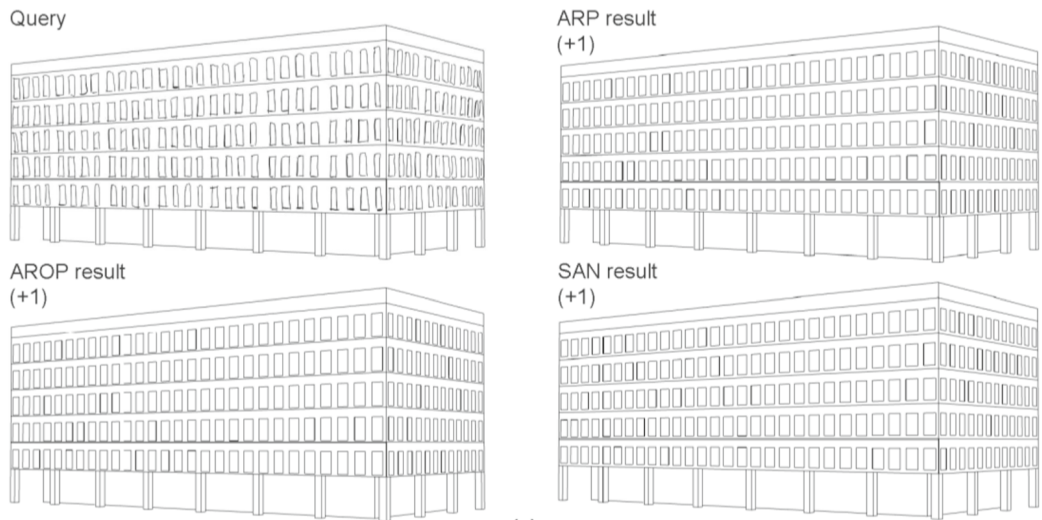
- Evaluating the effectiveness of all three algorithms.
- Evaluating AROP's effectiveness of using different template combinations (for setting pixels' orientations).
- Rating the engagement of using this method.
- Evaluating the usability of using this digital method.

In the first phase, respective top five images retrieved by three algorithms for 60 inquiry images, 900 images in total, are sent back to thirty participators who attended the workshop and made those inquiry images. Designers evaluate retrievals and give credits based on their own satisfaction degrees. To minimize influences of subjective biases, the satisfaction degree is only quantified by three scales: totally different from what I designed (− 1), very similar to what I designed (+ 1), between them (0). For example, all these three algorithms find satisfying images for the inquiry case illustrated in Fig. 8a, while all three algorithms retrieve unsatisfying images in the case of Fig. 8c. Between these two extremes, for majority design cases, always one algorithm is positively assessed while the other two are negatively or neutrally assessed, illustrated in Fig. 8b. In the end, the evaluation results of 60 cases are averaged to give out the final score. An algorithm is more satisfying when its final score is closer to '+ 1'. In addition, each algorithm's time consumption of image processing is also counted.

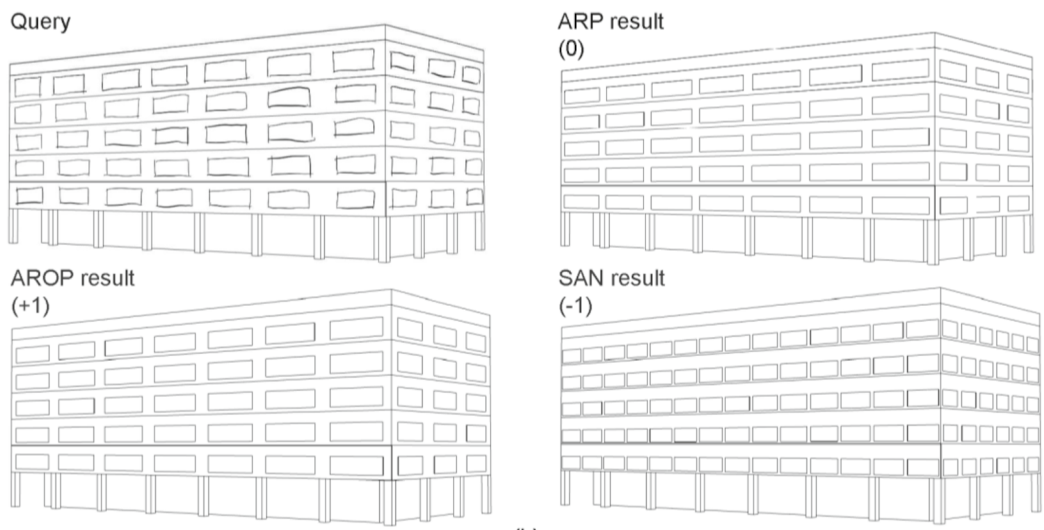
AROP with the h3v5 template combination has already been evaluated in the first phase. Therefore, in the second evaluation phase, only retrievals from using the other 15 template combinations are rated. At this time, participators only need to evaluate retrieval images ranked in the 1st position, with the same scoring system mentioned above.

In the third phase, designers need to evaluate retrievals answering incomplete inquiry images. The same operation as before, when one retrieval looks similar to what architect designed, one positive credit can be given to the algorithm. Otherwise, designers can give the algorithm zero credit, or minus one credit.

In the final evaluation phase, this SBIR decision-making method itself is also assessed by thirty architecture students who contributed to the inquiry dataset. They are asked to give opinions about the engagement level when trying this new method. Their engagement levels are measured basing on two questions with related credits. The first question is about 'interest': it is not interesting for me to design on the electronic screen and retrieve images (− 1), acceptable (0), it is interesting for me to design on the electronic screen and retrieve images (+ 1). Besides this, 'usability' as another aspect of engagement is also investigated: it is difficult for me to use this equipment (− 1), acceptable (0), I do not find the method complex or confusing at all (+ 1). In the end, evaluation scores from the 'interest' aspect and the 'usability' aspect are averaged to identify the total engagement level.


Query
ARP result
(+1)
AROP result
(+1)
SAN result
(+1)

(a)


Query
ARP result
(0)
AROP result
(+1)
SAN result
(-1)

(b)


Query
ARP result
(-1)
AROP result
(-1)
SAN result
(-1)

(c)

◂**Fig. 8** All retrievals from the three algorithms are positively evaluated (**a**); retrievals are differently evaluated (**b**); all retrievals from the three algorithms are negatively evaluated (**c**)

4 Results and discussions

4.1 The effectiveness of three algorithms

After the assessment for the initial phase, AROP is considered as performing the best among the three algorithms. As shown in Fig. 9a, its evaluation results reach 0.4 while the ARP results are only around zero. The results of SAN are below −0.5 and reach −0.6, much worse than the other two. This outcome means that except AROP, neither ARP nor SAN is competent to accomplish the current task. On the other side, as the image ranking going down, among all the three algorithms, no significant performance deterioration is detected. The reason is that many design solutions herein are quite similar. Therefore, images ranked in the 4th or 5th places still can satisfy designers' expectations.

Normally, deep-learning-based techniques perform better than geometric-feature based algorithms. However, SAN is a pre-trained CNN model used for sketch classification, while the task herein is to measure similarities. The CNN model is trained to recognize sketches from different classes, such as apples, tigers, and aeroplanes, etc. We can see that SAN is good at capturing the semantic differences between different sketch categories. Nevertheless, the sketches in this paper are much similar to each other. The difference between various inquiry and gallery images in this paper lies in the length of windows. For SAN focusing on dealing with the classification of different image categories, differences herein are too small to be recognized by its feature extractor. This is the reason SAN doesn't perform better. As for ARP and AROP, they focus on the number and orientations of the edge pixels. As a result, it suits the application of extracting features for design solutions, which makes the AROP scheme performs better than SAN.

Actually, the real performance of AROP could be even better. Although architects who made these inquiry images were informed of exact window wall ratios on facades, some of them still did not have a clear idea of how large these windows exactly should be. Therefore, for some query cases designed with too small or too large windows, there is no correct result can be retrieved from the gallery dataset, resulting in negative evaluations.

Besides the subjective satisfaction degree, time consumption is another important indicator to measure algorithms' usefulness. According to Table 3, which lists three algorithms' time consumptions, SAN consumes the least time to pre-process inquiry images and gallery images. AROP spends 4 s more than ARP on these tasks, due to the orientation information extraction. For the retrieving process, all three algorithms consume much less time, only about two milliseconds. In conclusion, from the perspective of time consumption, SAN is the best choice while AROP is the worst.

4.2 Results of the in-depth AROP investigation

4.2.1 Evaluation results of *h* and *v* combinations

Evaluation results of AROP using 16 different template combinations (Fig. 9b) identify that the majority of template combinations are positively evaluated, getting scores from 0.33 to 0.47. However, when a vertical template only uses two vertical pixels, no matter how many pixels are used for the horizontal template, retrievals become always negatively evaluated. The reason is that AROP in this study recognizes vertical pixels first, then recognizes horizontal pixels from the left ones. Therefore, when the number of pixels for a vertical template is too small, oblique lines are directly recognized as vertical ones because oblique lines usually have two or three consecutively vertical pixels. Therefore, these vertical templates with two pixels perform badly in the SBIR process.

On the other side, when a template combination recognizes vertical or horizontal lines with strict requirements, such as the combination 'h8v8', many vertical or horizontal lines are ignored by templates. Therefore, this kind of setting also influences retrieval results. As shown in Fig. 9b, template combinations with 8 pixels are not more satisfying than any other template combinations with fewer pixels. In conclusion, a template should not be set with too few nor too many pixels. For the current task, h5v8 is seen as the most satisfying combination.

4.2.2 Evaluation results of *M* and *N* combinations

As shown in evaluation results regarding combinations of M and N (Table 4), among 16 M & N combinations, the highest score (0.8) occurs when both M and N have their respective largest candidates, while the lowest score (0.42) appears when both M and N have the smallest choices. In general, the performance of AROP rises as values of M and N increase. When M and N grow, an image is divided into many smaller parts, which can help algorithms capture more tiny differences. In conclusion, in order to obtain a good performance, we should try to let M and N be large enough.

Nevertheless, when N is set to 4, 8 or 16, the retrieval performance does not always increase as M rises. When M is set to 8 or 16, some windows are split into two 'half windows', independently belonging to different image parts. These 'half windows' bring superfluous horizontal and vertical edge pixels, negatively impacting the algorithm

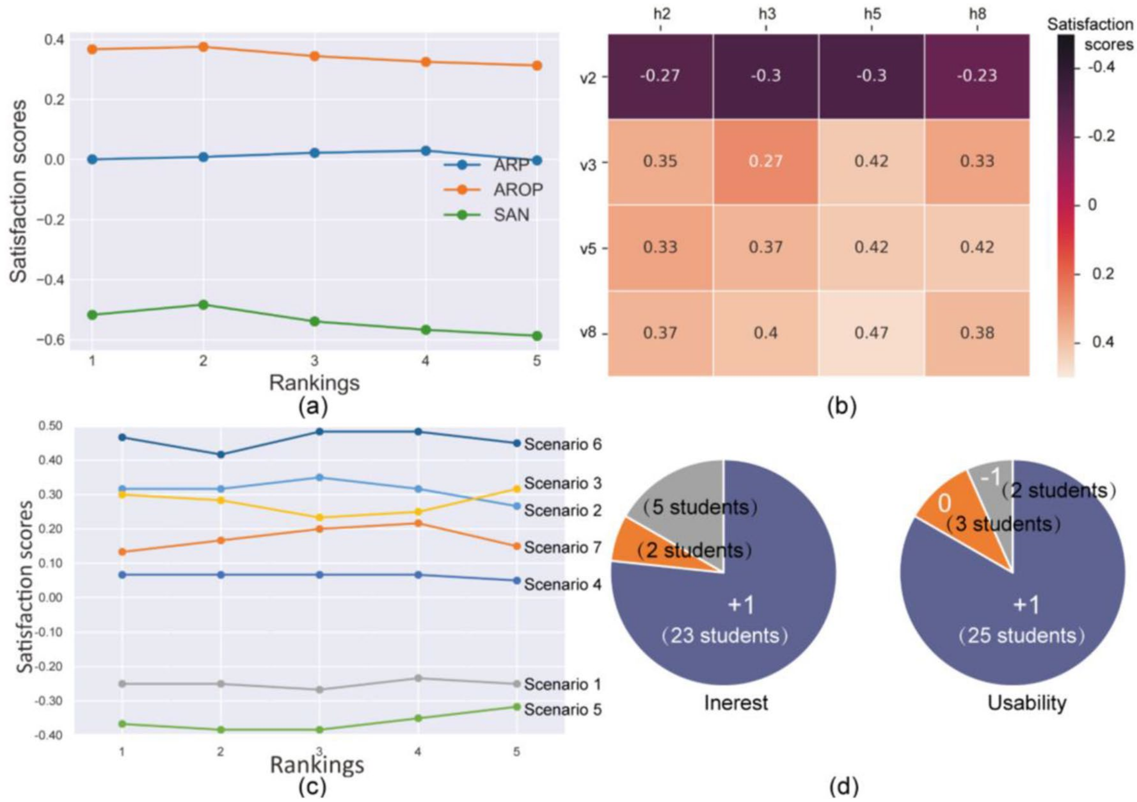


Fig. 9 Satisfaction degrees towards three algorithms (**a**); Satisfaction degrees towards AROP using different templates (**b**); Satisfaction degrees when using incomplete inquiry images (**c**); Results of engagement level investigation (**d**)

Table 3 The average time cost of pre-processing and SBIR for three algorithms

Algorithms	Data pre-processing of per inquiry image (s)	Data pre-processing of per gallery image (s)	SBIR of per inquiry image (ms)
ARP	1.778	1.763	2.45
AROP (h3v5)	5.200	5.837	2.13
SAN	0.323	0.273	2.17

Table 4 Evaluation results of different M and N combinations

	$M4$	$M8$	$M16$	$M32$
$N2$	0.42	0.53	0.53	0.60
$N4$	0.67	0.52	0.52	0.67
$N8$	0.70	0.60	0.65	0.75
$N16$	0.67	0.60	0.70	0.80

performance. However, when M is small or big enough, 'half windows' become less, and retrieval performance increases.

4.3 Results of answering incomplete inquiry images

Since AROP outperforms the other two algorithms in the previous step, it is also selected here to retrieve wanted images for incomplete inquiry images. There are seven experiments with seven different scenarios in total, and evaluation results are shown in Fig. 9c. Four scenarios which include more contents obtain average scores above zero, while scenario 1 and scenario 2 are below zero. Comparing four relatively better scenarios, it is hard to conclude any regular pattern from them, because scenario five, which contains the most drawn contents, does not perform the best. In short, evaluation results tend to be negative when the number of windows is not enough. However, when the number of windows reaches a bar, which kinds of strategies can go on raising the satisfaction score is still unclear.

Images from scenario seven have randomly drawn windows on building facades. Its retrieving results show that it outperforms scenario 1, scenario 4, and scenario 5.

Therefore, it is also possible that positions of drawn windows may also not influence the algorithm performance as expected. The authors also compare results of complete inquiry images and these images with randomly deleted windows. Among the total of 60 cases, there are 17 cases that complete inquiry image and incomplete inquiry with randomly deleted windows have the same 1st position retrieval.

The authors have to admit that the current AROP algorithm's performance still has the potentials to be improved. The underlying mechanism between incomplete inquiry images and retrievals deserves to be furtherly explored. For example, a scenario including more diversely located sections, a combination of section 1, section 4, and section 5, maybe more useful. Different from evaluation results toward the first experiment, which is used to compare three algorithms, ranking positions here nearly do not identify human's subjective perception at all. In several scenarios, retrievals ranked in position 4 and 5 receive higher scores than images in the 1st or 2nd position.

4.4 Results of designers' engagement level

The total credit identifying users' engagement level during the whole process is '0.7'. More details are shown in Fig. 9d. Although five students consider it is unacceptable to design on the LCD screen, and the other two students have a neutral attitude towards it, the left twenty-three architecture students enjoyed drawing with electronic devices, considering it is interesting. On the 'usability' side, apparently, it is more convenient and more advanced than traditional design methods.

Speaking of the time saved with the help of SBIR, it is also apparent that the time cost using traditional ways is totally not comparable with using SBIR. Half an hour is usually needed for architects to manually pick out the most similar image from the gallery dataset while algorithms can help them find it in seconds.

5 Research validation

5.1 Building Case 2

After finishing Building Case 1 investigation and its related algorithm research, the authors apply obtained research results to another case of building façade fenestration design (Fig. 10a) to validate those investigation findings. Case 2 is a high-rise office building with 27 floors (Fig. 10b). The ground floor is installed with glass curtains, and the other 26 floors need to be fenestrated. Unlike the previous building case, which has quite similar fenestration shape designs, more than one generative principle is applied to Building Case 2. Therefore, the window shapes of the second case are more diversifying. For example, windows in some cases do not only stay inside one singular floor, but they may cross two floors (Fig. 10c). Windows on the upper floor do not strictly align with windows on the lower floor. In total, for Building Case 2, there are 606 images in the gallery dataset.

Because of a much larger number of windows of Building Case 2, it is hard to draw every window on building facades. Therefore, inquiry images herein all have randomly drawn windows. In total, the authors prepare 30 inquiry images for Building Case 2. Meanwhile, the authors do not start the research from the beginning like what we did with Building Case 1, but directly taking AROP as the SBIR algorithm to be used herein, with the template of h5v5.

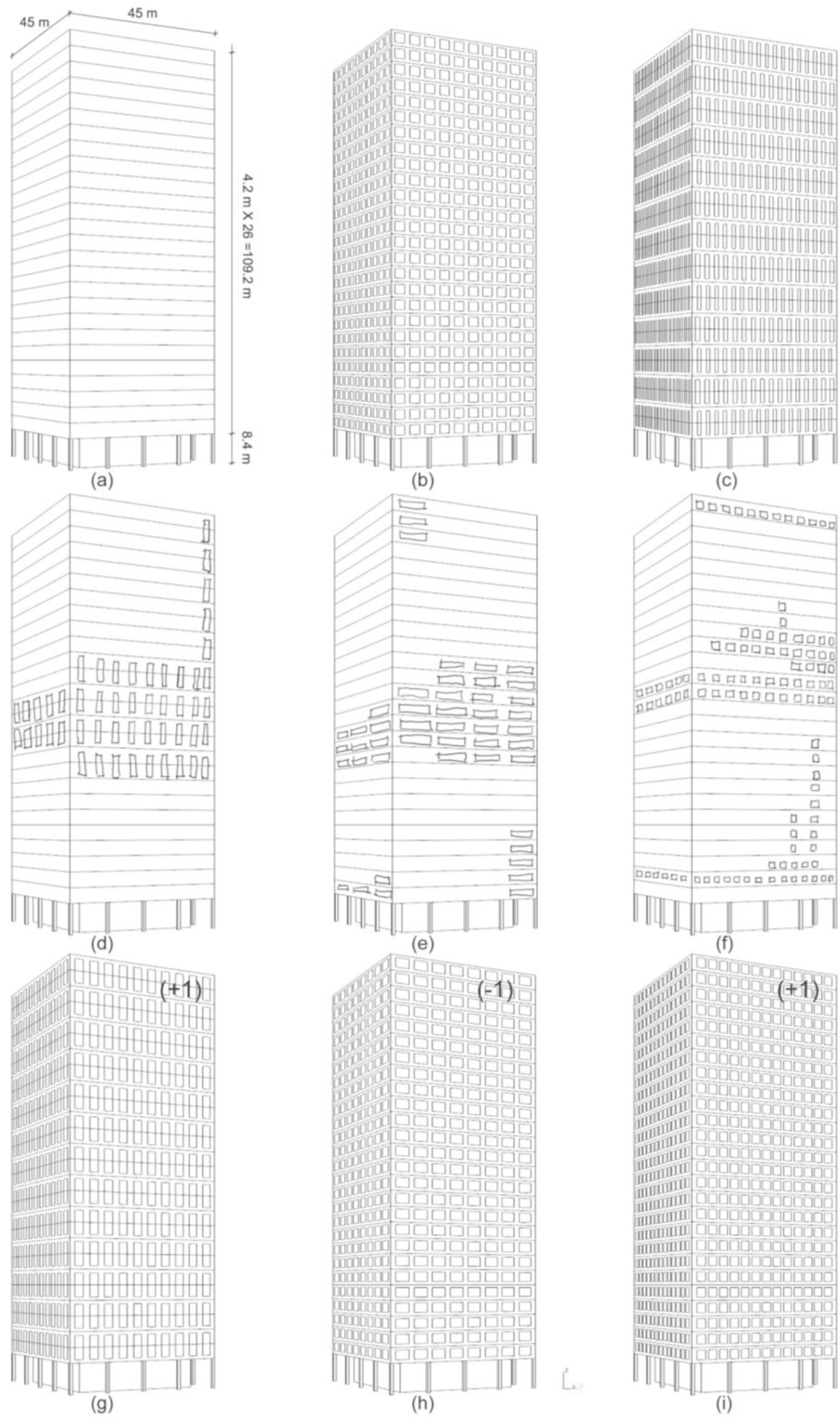
5.2 Results of the validation test

For Building Case 2, retrievals of some inquiry images (Fig. 10d–f) are illustrated in Fig. 10g–i. According to subjective evaluation, the average score of 30 retrievals achieves at 0.52. Therefore, we can say that the algorithm AROP, with obtained values of M & N combination, and h & v combination, can basically help designers find the most wanted gallery images, in the scenario of façade fenestration design.

6 Conclusions

PGAD is an emerging design method for high-performance buildings. However, real architectural design projects do not only need to consider building performances but also aesthetics, which is impossible to be solely pursued by computers. Architects and their clients still need to check the physical appearance of each design choice from the PGAD process. Therefore, even with some clustering software, the PGAD selection stage is still quite time-consuming and fatigued for architects. To solve this problem, the current study tries to automate the selecting process by integrating SBIR techniques, through which users can directly obtain the most aesthetically preferred design solution. This computer–human interaction speeds up designers' decision-making processes and incorporates their personal preferences within the decision process.

Three different SBIR algorithms are tested in this paper, and the authors find that AROP is the most suitable one here. A stepwise, regarding template combinations of AROP, which are used to decide pixels' orientations, the authors also have done several experiments and find that templates with 5 pixels are most useful. In the end, authors test incomplete inquiry images and find that the number of windows drawn on facades do influence the evaluation when they are not enough. Feedback of participators' engagement levels encourages some people, who doubt the application of


45 m
45 m
4.2 m X 26 =109.2 m
8.4 m
(a)
(b)
(c)
(d)
(e)
(f)
(+1)
(-1)
(+1)
(g)
(h)
(i)

◂**Fig. 10** The basic information of Building Case 2 (**a**); Two gallery images (**b, c**); Three incomplete inquiry images (**d–f**); Retrievals ranked in the 1st position for inquiry images def (**g–i**)

electronic devices for design, to use them in the architectural design field and explore new possibilities.

Four main research chances can be extended from the current study. Firstly, although the current study is only dealing with the case of fenestration design, this method actually can be applied to various applying scenarios, such as building shape design and floor layout design. Moreover, the future gallery datasets should be enriched by using as many various parametrization principles as possible, bringing in more geometric diversities and complexities for the SBIR process, adapting to real design projects.

Secondly, the current SBIR is implemented by our authors, not designers who lack programming skills. Therefore, when asking designers themselves to perform SBIR without a maturely designed GUI (graphical user interface), the usability of SBIR techniques may decrease. The authors plan to create a webpage as the front-end interface and put the whole calculation process on the cloud. It would be much faster and more user-friendly for common users without programming backgrounds.

Thirdly, SBIR actually can use different image features, such as colours [26]. Therefore, in future studies, the colour of window glasses can be distinguished from the other parts of building facades. That can help architects to design during the inquiry image drawing. Fifthly, the psychology mechanism, how humankind observes and compares two images, can be studied in the future.

Fourthly, unlike the other studies from the SBIR community, the degree of searching results' correctness herein is influenced by designers' subjective evaluation. Therefore, the small-scale evaluation in the current study may not completely identify algorithms' performances. The relationship between images and human cognition deserves to be furtherly explored.

Acknowledgements Ms. Jiewen WU contributes to plotting Fig. 9c.

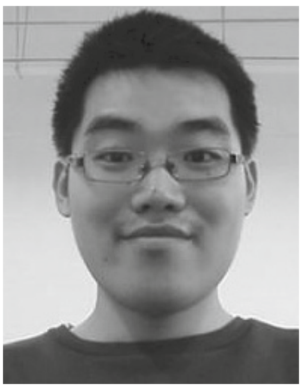
Funding Funding was provided by Key Technologies Research and Development Program (Grant No. 2020YFC2006602), National Natural Science Foundation of China (Grant No. 62072324) and Jiangsu Provincial Key Research and Development Program (Grant No. BE2020026).

References

1. Zhao, S., Angelis, D.E.: Performance-based generative architecture design: a review on design problem formulation and software utilization. J. Integr. Des. Process Sci. **22**(3), 55–76 (2019)
2. Gerber, D.J., Pantazis, E., Wang, A.: A multi-agent approach for performance based architecture: design exploring geometry, user, and environmental agencies in façades. Autom. Constr. **76**(April), 45–58 (2017)
3. Yang, D., Ren, S., Turrin, M., Sariyildiz, S., Sun, Y.: Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: a case of conceptual sports building design. Autom. Constr. **92**, 242–269 (2018)
4. Thornton, T.: Design Explorer 2. Github (2017). http://tt-acm.github.io/DesignExplorer/
5. Autodesk: Project Refinery Beta (2019)
6. Sileryte, R., Aquilio, A. D., Di Stefano, D., Yang, D., Turrin, M.: Supporting exploration of design alternatives using multivariate analysis algorithms. In: Proceedings of the Symposium on Simulation for Architecture and Urban Design (simAUD 2016), pp. 215–222 (2016)
7. Nagy, D., et al.: Project discover: an application of generative design for architectural space planning. SimAUD **2017**, 59–66 (2017)
8. Rodrigues, E., Sousa-Rodrigues, D., Teixeira de Sampayo, M., Gaspar, A.R., Gomes, Á., Henggeler Antunes, C.: Clustering of architectural floor plans: A comparison of shape representations. Autom. Constr. **80**, 48–65 (2017)
9. Liang, L.B., Jakubiec, J.A.: A three-part visualisation framework to navigate complex multi-objective (>3) building performance optimisation design space. In: BSO2018, pp. 11–12 (2018)
10. Suga, K., Kato, S., Hiyama, K.: Structural analysis of pareto-optimal solution sets for multi-objective optimization: an application to outer window design problems using multiple objective genetic algorithms. Build. Environ. **45**(5), 1144–1152 (2010)
11. Yousif, S., Yan, W.: Shape clustering using K-medoids in architectural form finding. In: Computer-Aided Architectural Design. ``Hello, Culture'', pp. 459–473 (2019)
12. Xiao, R.: Comparing and clustering residential layouts using a novel measure of grating difference. *Nexus Netw. J.* 0123456789 (2020)
13. Wallacei: Wallacei: an evolutionary and analytic engine for grasshopper 3D. https://www.wallacei.com/ (2019). Accessed 4 Apr, 2021
14. Yang, J., et al.: k-Shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement. Energy Build. **146**, 27–37 (2017)
15. Qian, X., Lu, D., Wang, Y., Zhu, L., Tang, Y.Y., Wang, M.: Image re-ranking based on topic diversity. IEEE Trans. Image Process. **26**(8), 3734–3747 (2017)
16. Wang, Y., Zhu, L., Qian, X., Han, J.: Joint hypergraph learning for tag-based image retrieval. IEEE Trans. Image Process. **27**(9), 4437–4451 (2018)
17. Wang, Y., et al.: Position focused attention network for image-text matching. In: IJCAI 2019 (2019)
18. Zhao, W., Xie, F., Zhao, W., Wang, X., Chen, L., Peng, J.: Tag-based weakly-supervised hashing for image retrieval. In: IJCAI Int. Jt. Conf. Artif. Intell., 2018-July, pp. 3776–3782 (2018)
19. Liu, P., Guo, J.M., Wu, C.Y., Cai, D.: Fusion of deep learning and compressed domain features for content-based image retrieval. IEEE Trans. Image Process. **26**(12), 5706–5717 (2017)
20. Zhu, L., Shen, J., Xie, L., Cheng, Z.: Unsupervised visual hashing with semantic assistant for content-based image retrieval. IEEE Trans. Knowl. Data Eng. **29**(2), 472–486 (2017)
21. Xia, Z., Wang, X., Zhang, L., Qin, Z., Sun, X., Ren, K.: A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. IEEE Trans. Inf. Forensics Secur. **11**(11), 2594–2608 (2016)
22. Hong, R., Li, L., Cai, J., Tao, D., Wang, M., Tian, Q.: Coherent semantic-visual indexing for large-scale image retrieval in the cloud. IEEE Trans. Image Process. **26**(9), 4128–4138 (2017)

23. Qian, X., Tan, X., Zhang, Y., Hong, R., Wang, M.: Enhancing sketch-based image retrieval by re-ranking and relevance feedback. IEEE Trans. Image Process. **25**(1), 195–208 (2016)
24. Zhang, Y., Qian, X., Tan, X., Han, J., Tang, Y.: Sketch-based image retrieval by salient contour reinforcement. IEEE Trans. Multimed. **18**(8), 1604–1615 (2016)
25. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: Sketch-based image retrieval: benchmark and bag-of-features descriptors. IEEE Trans. Vis. Comput. Graph. **17**(11), 1624–1636 (2011)
26. Bui, T., Collomosse, J.: Scalable Sketch-based image retrieval using color gradient features. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1012–1019 (2015)
27. Sun, X., Wang, C., Xu, C., Zhang, L.: Indexing billions of images for sketch-based retrieval. In: Proceedings of the 21st ACM international conference on Multimedia, pp. 233–242 (2013)
28. Parui, S., Mittal, A.: Similarity-invariant sketch-based image retrieval in large databases. In: Computer Vision—ECCV 2014, pp. 398–414 (2014)
29. Wang, L., Qian, X., Zhang, Y., Shen, J., Cao, X.: Enhancing sketch-based image retrieval by CNN semantic re-ranking. IEEE Trans. Cybern. 1–13 (2019)
30. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. ACM Trans. Graph. **35**(4), 1–12 (2016)
31. Liu, L., Shen, F., Shen, Y., Liu, X., Shao, L.: Deep sketch hashing: fast free-hand sketch-based image retrieval. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2862–2871 (2017)
32. Seddati, O., Dupont, S., Mahmoudi, S.: Quadruplet networks for sketch-based image retrieval. In: ICMR'17, pp. 184–191 (2017)
33. Qi, Y., Song, Y.Z., Zhang, H., Liu, J.: Sketch-based image retrieval via Siamese convolutional neural network. In: Proceedings—International Conference on Image Processing, ICIP, pp. 2460–2464 (2016)
34. Bui, T., Ribeiro, L., Ponti, M., Collomosse, J.: Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. Comput. Vis. Image Underst. **164**, 27–37 (2017)
35. Li, Y., Li, W.: A survey of sketch-based image retrieval. Mach. Vis. Appl. **29**(7), 1083–1100 (2018)
36. Xu, J., Xue, X., Wu, Y., Mao, X.: Matching a composite sketch to a photographed face using fused HOG and deep feature models. Vis. Comput. (2020)
37. Yu, Q., Yang, Y., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M.: Sketch-a-Net: a deep neural network that beats humans. Int. J. Comput. Vis. **122**(3), 411–425 (2017)
38. Takagi, H.: Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. Proc. IEEE **89**(9), 1275–1296 (2001)
39. Brintrup, A.M., Ramsden, J., Takagi, H., Tiwari, A.: Ergonomic chair design by fusing qualitative and quantitative criteria using interactive genetic algorithms. IEEE Trans. Evol. Comput. **12**(3), 343–354 (2008)
40. Muehlbauer, M., Burry, J., Song, A.: An aesthetic-based fitness measure and a framework for guidance of evolutionary design in architecture. In: International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar), vol. 12103, pp. 134–149. LNCS (2020)
41. Kowaliw, T., Dorin, A., McCormack, J.: Promoting creative design in interactive evolutionary computation. IEEE Trans. Evol. Comput. **16**(4), 523–536 (2012)
42. Weber, M., Langenhan, C., Roth-Berghofer, T., Liwicki, M., Dengel, A., Petzold, F.: a.SCatch: Semantic structure for architectural floor plan retrieval. In: ICCBR 2010: Case-Based Reasoning. Research and Development, pp. 510–524 (2010)
43. Langenhan, C., Weber, M., Liwicki, M., Petzold, F., Dengel, A.: Sketch-based methods for researching building layouts through the semantic fingerprint of architecture. In: Designing Together: CAADFutures 2011—Proceedings of the 14th International conference on Computer Aided Architectural Design, pp. 85–102 (2011)
44. Langenhan, C., Weber, M., Liwicki, M., Petzold, F., Dengel, A.: Graph-based retrieval of building information models for supporting the early design stages. Adv. Eng. Inform. **27**(4), 413–426 (2013)
45. Ahmed, S., Weber, M., Liwicki, M., Langenhan, C., Dengel, A., Petzold, F.: Automatic analysis and sketch-based retrieval of architectural floor plans. Pattern Recognit. Lett. **35**(1), 91–100 (2014)
46. Roith, J., Langenhan, C., Petzold, F.: Supporting the building design process with graph-based methods using centrally coordinated federated databases. Vis. Eng. **5**(1) (2017).
47. Kai, Q.: Drawing semantic retrieval algorithms based on deep multilayer convolutional network. In: Proceedings—2019 International Conference on Smart Grid and Electrical Automation, ICSGEA 2019, pp. 255–258 (2019)
48. Wessel, R., Ochmann, S., Vock, R., Blümel, I., Klein, R.: Efficient retrieval of 3D building models using embeddings of attributed subgraphs. In: International Conference on Information and Knowledge Management, Proceedings, pp. 2097–2100 (2011)
49. Wessel, R.: Shape Retrieval Methods for Architectural 3D Models. Universität Bonn (2013)
50. Yasseen, Z., Verroust-Blondet, A., Nasri, A.: View selection for sketch-based 3D model retrieval using visual part shape description. Vis. Comput. **33**(5), 565–583 (2017)
51. Kazi, R.H., Grossman, T., Cheong, H., Hashemi, A., Fitzmaurice, G.: DreamSketch: early stage 3D design explorations with sketching and generative design. In: UIST 2017—Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, pp. 401–414 (2017)
52. Zhao, S.: Searching for office buildings' fenestration geometries with a bi-phase optimization framework. Sci. Technol. Built Environ. **20**(9), 1–22 (2020)
53. Roudsari M.S.: Honeybee. Github (2019)
54. Jonatan: TT TOOLBOX. *food4rhino* (2017). https://www.food4rhino.com/app/tt-toolbox
55. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-8**(6), 679–698 (1986)
56. Martin, D.R., Fowlkes, C.C.: Learning to detect natural image boundaries using local brightness, color, and texture cues. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 5, pp. 530–549 (2004)
57. Chalechale, A., Mertins, A., Naghdy, G.: Edge image description using angular radial partitioning. IEE Proc. Vis. Image Signal Process. **151**(2), 93–101 (2004)
58. Yu, Q., Yang, Y., Song, Y.-Z., Xiang, T., Hospedales, T.: Sketch-a-Net that beats humans (2015)
59. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)

Shenghuan Zhao obtained his Master's and Ph.D. degrees from Politecnico di Milano, Italy. Currently, he works as an assistant professor at Suzhou University of Science and Technology, China. His main research interests include algorithmic optimization for building industry, IoT-based built environment, interactive design in AR and MR.



Luo Wang is currently pursuing the Ph.D. degree with SMILES Lab, Xi'an Jiaotong University, Xi'an, China. His current research interests include sketch-based image retrieval, image content understanding, and deep learning.

Xueming Qian (M'09) received the B.S. degree in industrial automation and the M.S. degree in pattern recognition from Xi'an University of Technology in 1999 and 2004, respectively, and Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, in 2008. From 2011 to 2014, he was an Associate Professor with Xi'an Jiaotong University, where he is currently a Full Professor and the Director of SMILES Lab. He was a Visiting Scholar with Microsoft Research Asia, Beijing, China, from 2010 to 2011. His current research interests include social media big data mining and search. Prof. Qian was a recipient of the Microsoft Fellowship in 2006 and the Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province in 2010 and 2011, respectively.

Jianping Chen is a professor at Suzhou University of Science and Technology, China. He received the Bachelor's degree from Southeast University, China, the Master's degree from New Jersey Institute of Technology, USA, the MBA from Washington University in St. Louis, USA, and the Doctor's degree from University of Nice Sophia Antipolis, France. His research interests include big data and analytics, building energy efficiency, and machine learning.