IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 December 2023, accepted 29 January 2024, date of publication 2 February 2024, date of current version 8 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3361659

RESEARCH ARTICLE

DualShape: Sketch-Based 3D Shape Design With Part Generation and Retrieval

XUSHENG DU, TIANYU ZHANG, AND HAORAN XIE, (Member, IEEE)
Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923-1211, Japan

Corresponding author: Haoran Xie (xie@jaist.ac.jp)

This work was supported in part by the Japan Society for the Promotion of Science Grants-in-Aid for Scientific Research (JSPS KAKENHI) under Grant 23K18514, and in part by the Kayamori Foundation of Informational Science Advancement.

ABSTRACT Creating a 3D shape design from free-hand sketches is a challenging task due to the sparse and ambiguous information from sketches. In this work, we propose DualShape, a sketch-based 3D shape design interface with part generation and retrieval. DualShape first decouples the model generation into two parts: a composite part retrieval module using a sketch-based feature matching method and a sketch-based part generation module using a deep learning approach with implicit function representation. We then propose an assembly module for the obtained part models to accomplish the task of 3D shape generation from input sketches. In addition, we provide an optimization module that allows users to optimize and manually adjust the assembled model to achieve satisfying models. For the reconstruction interface, shadow guidance was utilized to assist users with the retrieved 3D models matching the input strokes as background references in real time. To verify the effectiveness of the DualShape system, we conducted the comparison experiments and the user study. The results show that DualShape is more user-friendly and is able to generate 3D models with richer details. We believe this work would provide a novel paradigm for hybrid 3D model generation in computer graphics.

INDEX TERMS Sketch-based interface, shape generation, shadow guidance, user interface, part assembly.

I. INTRODUCTION

3D shape reconstruction is a significant and challenging research topic in computer graphics, that has been widely adopted in various applications such as entertainment and architecture design. 3D modeling with conventional commercial tools like Autodesk, AutoCAD, and 3ds Max is usually laborious and requires professional knowledge and specialized skills. One feasible solution to easily obtain a 3D model is to capture and scan objects with scanning devices, but the special devices are usually expensive and may have difficulty reconstructing complex and detailed geometries. Recently, deep learning-based approaches have emerged for 3D model creation [6]. However, a large amount of data is required to train the learning model, and the generated results may be unsatisfactory due to inadequate training data and the strategy of generative models. In this work, we aim to provide a hybrid approach combining shape retrieval and shape generation for model design tasks.

The motivation of this research is to support users in designing models from free-hand sketches without professional skills. The sketch-based methods for 3D model reconstruction and designs can simplify the modeling process with intuitive controls [1]. However, these methods may have difficulty with inaccurate and sparse sketch inputs. To solve this issue, it remains challenging to guide users performing sketches for 3D modeling. User guidance with built-in templates and pre-designed objects may require a significant amount of extra time to modify models to achieve satisfying results. While guiding through tutorials and step-by-step instructions, the users cannot directly conduct modifications in the modeling process. In this work, we provide a shadow guidance method that guides users to draw sketches in real time to assist in 3D model generation.

The associate editor coordinating the review of this manuscript and approving it for publication was Giambattista Gruosso.

It is difficult to generate and design 3D models interactively with fine details. Previous generative models like MeshSDF [31] fail to achieve this goal, as shown in Fig. 1. Take a 3D car model as an example, firstly, since a car shell is composed of various parts such as the front cover, roof, rear cover, doors, and frame, it is challenging and tedious for common users to design these parts in detail with lots of manual operations. Considering the complexity and diversity of car shells, we observed that the generation approach to generating the component models from sketch input is more feasible than retrieval-based approaches. Secondly, the four tires of a car are consistent with limited patterns and symmetrical structures. We observed that the sketching and generation of the four tires as the entire tire component may reduce the effort of drawing these tires individually. However, compared with the car shell, the car tires are more detailed and have smaller sizes (as shown in Fig. 1(c)), and are more difficult for users to draw. Therefore, the retrieval-based approaches may obtain fine-detailed models for car tires even using rough sketches. Moreover, the distinction between the parts (e.g., the connection between the car shell and the car tires) is usually clear, as shown in Fig. 1. With the help of the assembly approach, it becomes possible to generate a model with a clear connection structure, as depicted in Fig. 1(a).

In this research, we propose DualShape, a novel 3D shape design framework using hybrid shape representations, as shown in Fig. 2. The proposed system uses user-drawn sketches as input, combining the advantages of the implicit representation of 3D shape as a signed distance function (SDF) [29] and the explicit representation using the shape retrieval approach. Therefore, we perform the task of 3D model generation based on SDF and also utilize a data-driven retrieval approach to obtain component models in the design process. In addition, to assist users in drawing sketches corresponding to the model's characteristics, the design interface can retrieve multiple models that match the characteristics of the user's sketch in real time as shadow guidance. We conducted the comparison experiments and the user study to evaluate DualShape from objective and subjective perspectives, respectively. The results of the comparative experiments demonstrate that our system can generate models with richer details. The results of the user study show that our system is easy to use and user-friendly. Most of the users were satisfied with the generated results.

The main contributions of this work are listed as follows:

- A novel design framework for 3D models that takes a hybrid approach with the generative model of implicit shape representation and shape-based shape retrieval.
- With a data-driven approach, we adopt the decomposition and assembly for the design of individual 3D model parts based on their geometric features and design tasks.
- An interactive user interface based on the proposed framework with user design guidance ensures that common users can design 3D models efficiently, even for novices.

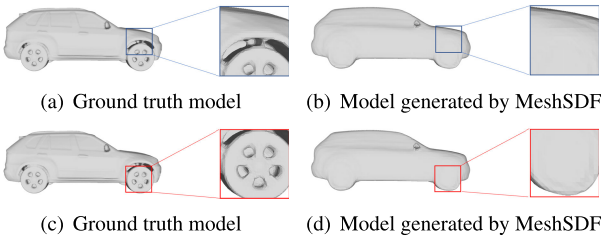


FIGURE 1. The ground truth model and the model generated in MeshSDF [31]. (a) The structure of the connections between the parts of the ground truth model is clear. (b) The connection structure between parts is blurred in the model generated by MeshSDF. (c) The ground truth model of the car tire parts has a clear structure and distinct features. (d) The detailed features of the car tires are blurred in the model generated by MeshSDF.

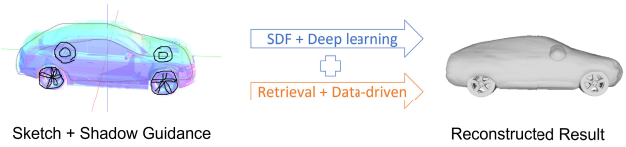


FIGURE 2. The proposed DualShape adopts the hybrid 3D model generation with part retrieval and generation.

II. RELATED WORK

A. IMPLICIT SURFACE REPRESENTATION

The implicit representation of the 3D model describes a model surface with a zero-crossing point of a volume function [32]. The advantage of this implicit representation is that the surface of the zero point can change the topological structure without explicit reparameterization. Representing a 3D shape as a set of level sets of a deep neural network and mapping 3D coordinates to a signed distance function [29] or occupancy field [24] can create a lightweight, continuous shape representation with no resolution limits. However, one major drawback of implicit representations is that they require active sampling and querying of 3D coordinates to construct the surface. For applications that require explicit surface parameterization, the non-differentiability of the standard level set extraction method [21] remains a barrier to leveraging the merits of implicit representations. The MeshSDF [31] recently overcame this problem by proposing a differentiable method for generating explicit surface mesh representations from signed distance functions. In this work, we aim to adopt the signed distance function to generate the 3D model.

B. SKETCH-BASED MODEL RETRIEVAL AND MODELING

Sketch-based 3D object retrieval and modeling have been explored extensively in the field of computer graphics [2]. Loffler et al. [20] introduced a system that allows users to refine keyword-based initial searches using sketches of the desired views. Meanwhile, Funkhouser et al. [7] proposed

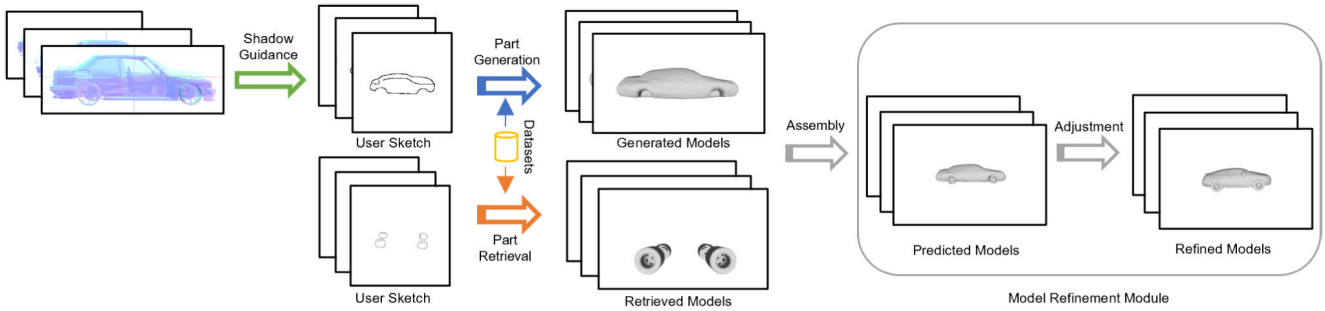


FIGURE 3. Workflow of the proposed DualShape which takes the hand-drawn sketches as input. After obtaining the corresponding part models through the retrieval and generation modules, the part models are assembled into a complete model. Finally, users can perform optimization adjustments based on the assembled model and output the refined result.

an image-based retrieval method. The shape search engine they provide is an early example of a 3D search service. It contains a drawing interface that allows users to upload the projections of a shape. Ma et al. [23] proposed a sketch-based retrieval method that uses stroke features detected by dense sampling points on the stroke. Eitz et al. [5] used the BoF feature function for sketch-based image retrieval. In this work, we construct sketch-based shape retrieval based on instance parts with the sketch features of the individual parts rather than the whole object.

Compared to images with rich information, sketches are often sparse and ambiguous. Igarashi et al. [14] introduced an interactive drawing system that can generate 3D shapes automatically in real time from 2D contours drawn by the user. Recently, deep learning-based methods have been proposed for the sketch-based modeling task. Lun et al. [22] and Li et al. [17] suggested first converting the input sketches to depth/normal maps and fusing them to construct a complete 3D model. However, all these methods were performed based on overall sketches, resulting in missing details for objects. Han et al. [9] utilized deep neural networks to predict potential codes of faces from 2D sketches to generate detailed face models, while Nishida et al. [27] trained networks to predict program model parameters to produce detailed shapes from sketches. These methods can produce complex, high-resolution models, but only for shapes that can be generated programmatically. To handle more general shapes, Delanoy et al. [1] designed an end-to-end convolutional neural network to generate the corresponding 3D shapes in a volume-based representation of a given input sketch. However, this approach is limited to low-resolution 3D meshes. In this work, we propose a hybrid method to perform the model generation task by combining two components: a data-driven retrieval method utilized for parts with high-quality details, and a generation approach applied for the general parts.

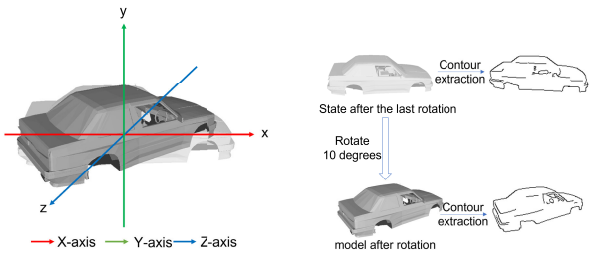
Commercial modeling tools like Maya and 3ds Max allow users to design objects using part-based modeling and assembly, which are difficult for non-experts. For assembly-based modeling, Li et al. [18] proposed learning part generation and assembly for structural shape generation based on volume representation; however, their method is based on semantics and is not suitable for recovering shapes with complex structures. To break out of the issue, Du et al. [3] decomposed the generation task into modeling and shape assembly based on parts. Based on the shape structure learning [25], [28] and the densely partitioned dataset [26]. In this work, we also divide the artificial objects into instance parts and then perform sketch-based retrieval and generation to improve the modeling effectiveness.

C. SKETCH GUIDANCE

Sketch guidance is used to guide users to update strokes by providing drawing feedback. The interactive sketching interfaces, such as Teddy [14] and the 3D drawing system [13], were designed to help users create better drawings. However, these methods only offer low-level feedback on information in terms of the basic shapes of lines, curves, and polygons. The shadow guidance can be calculated based on retrieving actual images from an image repository [16]. It then merges the images retrieved as shadows for the guidance of the drawing. Shadow guidance has been used widely in sketch-based applications, such as portrait drawing [10], anime creation [11], and motion retrieval [30]. Limpaecher et al. [19] automatically corrected the user drawings to draw variations of the same target subject in real time based on previously collected data. Iarussi et al. [12] provided a set of construction lines as visual guidelines to assist users. Many current research models assist users in drawing sketches from the image perspective. In this work, we used blended 3D models as shadow guidance to help users intuitively understand the features and structures of the model.

III. PROPOSED FRAMEWORK

In this work, we propose DualShape, a hybrid 3D shape design framework based on the sketch. As shown in Fig. 3, our framework is mainly composed of four modules: the part retrieval module; the part generation module; the part assembly module and the model refinement module.



(a) Model state in coordinate system (b) Contours extraction

FIGURE 4. Example of rotating the model to extract contours.

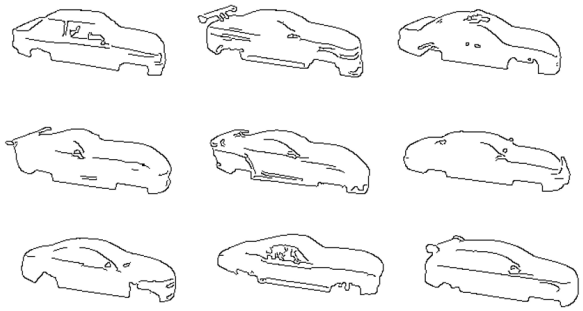
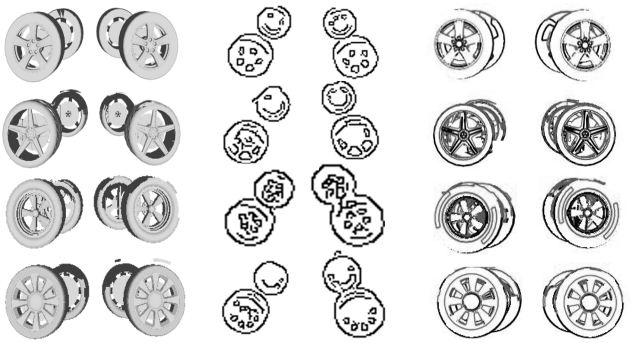


FIGURE 5. Examples from the car shell sketch dataset.



(a) car tire (b) tire contour(canny) (c) tire contour(openSSE)

FIGURE 6. The tire model and the contour images extracted by different methods: (a) the tire model, (b) the contour of the corresponding model extracted using the OpenCV's canny edge detection method, and (c) the contour of the corresponding model extracted using the OpenSSE method.

A. DATASETS

We adopted the approach of assembling complete models based on instance parts. Because there is no suitable dataset that meets our research purpose, we constructed both part and contour datasets by partitioning the models according to the part labels and extracting the contour images from each instance part.

We observed that it is meaningful to split the 3D model into structurally instance parts rather than the trivial small components. Taking the car category as an example in the dataset, a 3D car model can be split into two structural instance parts, the car shell and the car tyres. The clear hierarchical structure can make the model generation more concise and reduce the complexity of the 3D model. In addition, splitting the model into two main parts makes it easier to reuse these parts in other scenarios and applications. Overall, splitting the car model into two instance models, the shell and the tyres, can improve the maintainability and reusability of the models.

1) PART DATASET

The part dataset was constructed based on the ShapePFCN dataset [15], where 3D models were collected ''in the wild''. The car models were segmented into 4 labels (roof, hood, frame, and wheels). In this work, we split the models into two instance parts to simplify the user's drawing process: the car shell (made up of the roof, front cover, rear cover, and frame) and the tires. These instance parts were saved separately as model files. Thus, the part dataset has 1000 model instances, 500 car shells, and 500 tires.

2) PART CONTOUR DATASET

As the sketch input from the user interface, it is necessary to extract the model's contour data based on the obtained instance parts. For the part generation, considering the rough and concise situation of the lines drawn by the user, we extracted the contours of the car shell through canny edge detection. The extracted model contour is close to the user's hand drawing. As shown in Fig. 4, for extracting the car shell contour, we have custom-set the model with the Y-axis in 3D (x,y,z) coordinates as the center axis, and the rotation angle is 10 degrees each time; therefore, we can obtain the contour of each model at 36 viewpoints. In this study, 18,000 contours of car shells were collected in our dataset. Examples of car shell contour data are shown in Fig. 5.

For the part retrieval module, due to the data quality of the split wheel tires, canny edge detection can not accurately represent the characteristics of the tires. In addition, the differences among contour lines of tires are small, and the feature differences are not obvious. It is difficult to obtain the correct model during the retrieval process. Fig. 6(b) shows the tire model and the contour map obtained using the canny method. To address the issue of feature blurring when extracting tire contours, We adopt the open-source sketch search engine for 3D object retrieval (OpenSSE) [4], [33] (Fig. 6(c)). 102 viewpoint matrices are acquired from 102 uniformly distributed viewpoint directions and used to capture the contour images of each model. In our dataset, 51,000 contour images were collected. The contour images obtained by the OpenSSE method were more explicit in representing the characteristics of the tires.

B. PART RETRIEVAL

To simplify the process of generating the entire 3D model of cars, we utilized a data-driven retrieval-based approach based on the symmetric structure of the overall tires and the repetitive features of the tire pattern. Users can draw the tire pattern to retrieve a matching model and apply the result

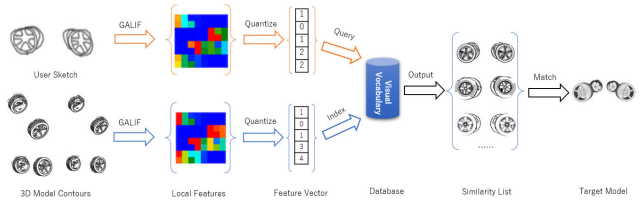


FIGURE 7. Part retrieval module. By extracting the features of the sketch drawn by the user through GALIF, querying the data with a high match in the constructed visual vocabulary, and using the model of the highest similarity data as the target part for retrieval.

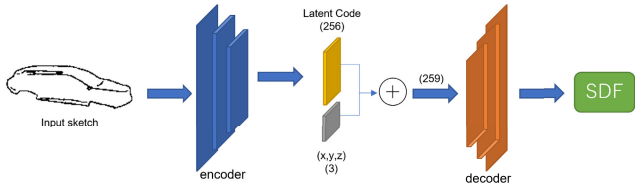


FIGURE 8. Network structure.

to the complete model. Fig. 7 illustrates the basic functions of the part retrieval module. Our shape retrieval method mainly includes two parts: (1) coding the line drawing with the bag-of-features (BoF) method to extract the features of the contour database, and constructing the visual vocabulary table; (2) extracting the features of the user-drawn sketches, and searching according to the features. The BoF method was utilized to extract features from 51,000 images in the tire contour dataset. By randomly selecting interest points in the images, the gradient direction histogram was calculated for each interest point's neighborhood. The main direction was taken as the edge feature of the interest points, ultimately forming a feature bag. A GALIF filter is applied to extract the local features of these images.

The sketch drawn by the users is first processed to extract the local features, and the features are converted into a feature vector of smaller dimensionality. During the retrieval process, the closest match of this vector is found in the visual vocabulary constructed in advance. For the whole image, multiple retrieved images with different degrees of similarity can be obtained by calculating the occurrences of cluster matches. We obtain the corresponding tire model based on the image with the highest similarity and load this model into the user interface. If the result does not meet expectations, users can redraw the sketch and repeat the retrieval work until a satisfactory tire model is obtained.

C. PART GENERATION

It is difficult to summarize and generalize the characteristics of car shells as composite parts. With the help of implicit fields of 3D shapes, such as SDF, geometric deep learning methods allow for detailed modeling of surfaces with any topological structure without relying on 3D Euclidean grids, resulting in learnable resolution-independent parametrization. In this work, we adopted a differentiable method to generate an explicit surface mesh representation from SDF as MeshSDF [31]. Fig. 8 shows the network structure of our generated components. We describe modeling shapes as the zero iso-surface decision boundaries of the network trained to represent SDFs. For a given spatial point x, the SDF function for its distance to the closest surface is defined as follows:

$$SDF(x) = s : x \in \mathbb{R}^3, s \in \mathbb{R} \tag{1}$$

The underlying surface is implicitly represented by the surface of $SDF(\cdot) = 0$. Given a sketch as input, it is expected that the network can generate a suitable SDF, and we can define the objective function as equation 2.

$$SDF = D(E(S), G) \tag{2}$$

Here, S denotes the input sketch, E defines the encoder that encodes the input sketch into latent code z_s. G is the set of features with the coordinates of the sampled points. The latent code z_s and features G are concatenated as latent vector z input to the decoder D to generate the SDF. The decoder D predicts the signed distance d_i for all coordinates $p_i \in G$.

In our implementation, the encoder is similar to MeshSDF [31]. It is composed of a ResNet18 network, and the input sketch is conditioned with a depth-implicit field by a residual image encoder that maps the input sketch to a latent code vector. In addition, the construction of the auto-encoder in DeepSDF [29] was utilized. The latent code vectors obtained by the encoder were used to condition the multilayer perceptron (MLP) architecture with the SDF.

D. PART ASSEMBLY

The retrieval and generation of parts only focus on retrieving/generating appropriate 3D shapes without considering global information. Since the obtained model does not consider the placement, all object parts are modeled uniformly in the center of the system (as shown in Fig. 9(a)), which does not meet the objective of generating the model. We estimated the distance and positional relationship between the complete car models and their parts in the dataset, and calculated the positional and proportionality relationships between the car shells and tires according to the geometric center point.

In the initial state of the assembly part, to achieve a reasonable placement and splicing of car shells and tires, we defined a set of rules based on prior knowledge that estimates the location and proportional relationships from the inter-model. The designated rules were set as follows:

- **Keeping center alignment.** For car shells A and tires B, the x and z values in the coordinates (x, y, z) of each part are set to 0 by default for quickly assembly. Only the y coordinate values must be adjusted according to the overlap to assembly. Fig. 9(a) illustrates the rule for keeping the center alignment. It shows that the centers of the bounding boxes of both the car shell and tire models

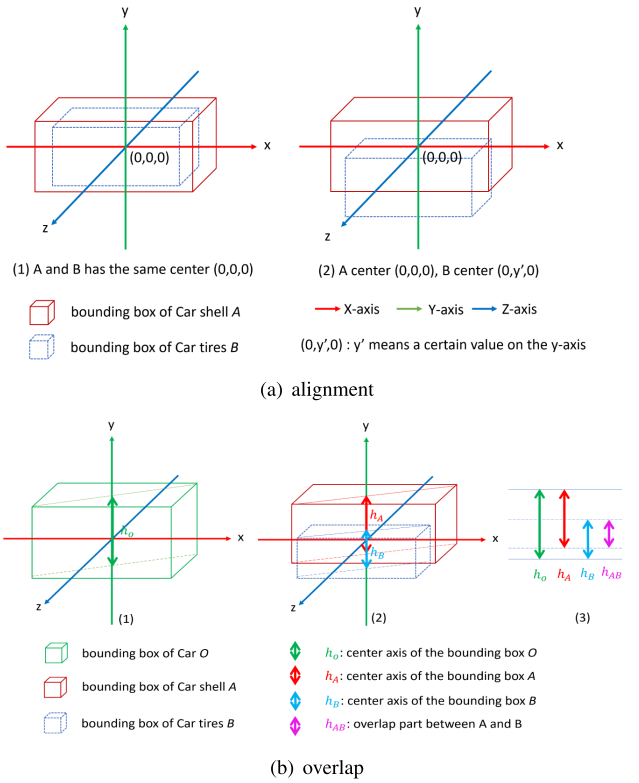


FIGURE 9. The basic rule for keeping the center of the model in alignment (a), and for maintaining a certain overlap ratio between part models (b).

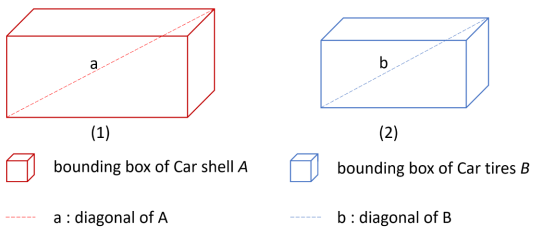


FIGURE 10. The basic rules for maintaining proportionality between part models.

are at the same position, i.e., the origin of the coordinate axis (0, 0, 0).

- **Maintaining proportionality.** There are proportionality relationships between car shells A and tires B, usually, the car shells are larger in scale than the tires. By calculating and maintaining fixed proportionality, the size of one part can be adjusted according to the fixed size of another part. Fig. 10 shows the proportional ratio between two-part models. We set a as the diagonal of the bounding box of model A and b for model B. The proportional relationship between the aspect ratios of the two models can be equated to the proportional relationship between the diagonals of the models' bounding boxes. Note that the center of the bounding box for the car shell is the origin of the coordinate, the center of the bounding box for the tire is $(0, y', 0)$, where y' is a specific value on the y-axis indicating that the tire's can be moved along the y-axis.
  Specifically, the proportional relationship between the models is calculated by comparing the diagonal lengths of the bounding box of the models. The calculation of the proportion relationship is defined as follows:

$$s_r = \frac{l(b)}{l(a)} \tag{3}$$

where s_r represents the scale ratio between the part models. The function l is defined to calculate the diagonal length of the model's bounding box:

$$l = \sqrt{l_x^2 + l_y^2 + l_z^2} \tag{4}$$

where l_x, l_y, and l_z represent the differences in the corresponding vertex coordinates of the bounding box on the three coordinate axes.

- **Maintain a certain overlap ratio.** Relative to the complete car model O, the bounding box of car shells A and tires B assembled together have a certain overlap part. A fixed overlap ratio allows the parts to be placed in the corresponding positions. Fig. 9(b) shows the overlapping relationship between parts when they are assembled into a complete model. According to fixed proportionality, the overlapping relationship is simplified from calculating the overlapped volume to calculating the overlapped height. Specifically, Fig. 9(b)-(2) shows that when car shells A and tires B are assembled into a complete model O, there will be a fixed height overlap between parts A and B, i.e., h_{AB}. Fig. 9(b)-(3) shows that the height of the overlapped parts needs to maintain a certain proportional relationship with the height of the complete model, and the proportional relationship is expressed as the following equation 5:

$$o_r = h \cdot \frac{h_{AB}}{h_O} - \frac{1}{h_O} \tag{5}$$

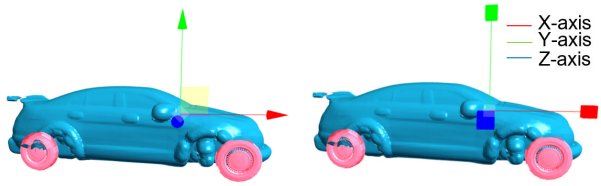
where o_r indicates the overlap ratio of the overlapping height in the height of the complete model. h_{AB} denotes the height of the overlapping part between the two models, h_O represents the height of the entire overlapping area, and h represents the maximum height of the two models (i.e., the height of the tallest point in either model).

E. MODEL REFINEMENT

After completing the assembly stage, DualShape also helps users adjust the assembly results freely if they are dissatisfied with them. As shown in Fig. 11, it mainly includes two functions: adjusting the model's position and adjusting the model's scale size. This enables the user to perform manual adjustments to obtain satisfactory results.

IV. USER INTERFACE

The interface of the DualShape system mainly consists of four parts: the shadow guidance part, the sketch operation



(a) Translation state (b) Scaling state

FIGURE 11. Two operation states when adjusting the model. (a) shows the operation state when translating the tire model, and (b) shows the operation state when scaling the tire model.

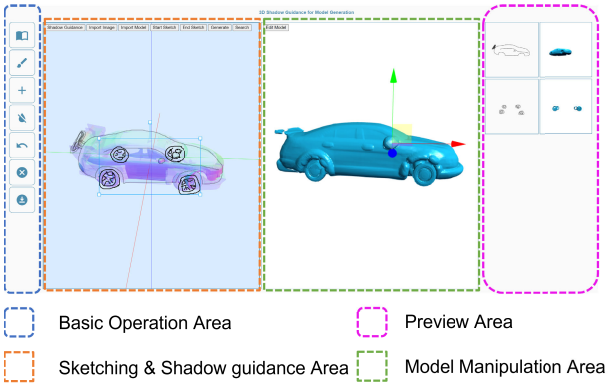


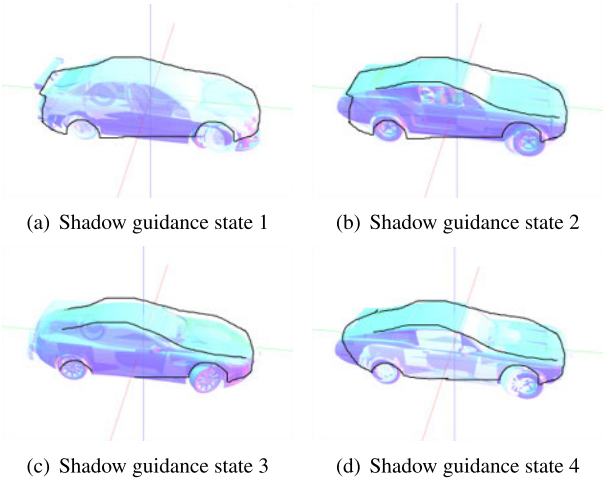
FIGURE 12. Overview of the user interface which is divided into 4 main areas: (a) the basic operation area, which contains functions such as drawing/deleting/downloading; (b) the drawing area, which is also the display area of the background model; (c) the display and operation area of the model, where the user can choose to enter the editing state and adjust the model details; and (d) the preview area, which displays the sketch of each part and its corresponding model.

part, the preview part, and the assembly part (Fig. 12). In the shadow guidance part, DualShape can retrieve and combine multiple models with similar contours in real time to provide a reference to users. In the sketch operation part, users can interact with the front-end system running in the browser to create and draw the sketches of each part. In the preview part, users can view the current models retrieved/generated based on the input sketches. In the assembly, users can further adjust the details based on the automatic assembly of parts.

A. SHADOW GUIDANCE

We implemented 3D object retrieval system based on contour feature lines as input [5]. Especially, we utilized an open source sketch search engine OpenSSE [33] for 3D object retrieval based on sketch images as input. In addition, the function of shadow guidance was implemented based on the retrieved results.

DualShape uses 3D models as background references to assist users in drawing model sketches. Specifically, DualShape provides a background template to retrieve similar models based on users' sketches in real time. When the user lifts the brush at the end of the sketch, we transfer the current sketch as input into the search engine, which extracts the sketch features and matches them with the features in the contour database to obtain multiple approximate models.



(a) Shadow guidance state 1 (b) Shadow guidance state 2

(c) Shadow guidance state 3 (d) Shadow guidance state 4

FIGURE 13. With the changes of the user sketch, the 3D model used as a background is also updated in real time. When strokes are added on top of the sketch in (a), the model used as shadow guidance also changes to the state of (b). When strokes are deleted from the base of the sketch in (b), the 3D model as shadow guidance also changes from (b) to (c) accordingly. Adding strokes to the sketch of (c) after deleting strokes, the shading guidance models are also changed to (d).

The obtained multiple models are composited according to the transparency ratio, which can provide a shadow-guided 3D model as output. The process can be represented by the following formula:

$$M_{\text{output}} = \sum_{i=1}^{n} \alpha_i \cdot M_i \tag{6}$$

Here, M_{output} is the final output 3D model, representing the result obtained after processing the user-drawn sketch. M_i is the i-th approximate model retrieved by the search engine. α_i represents the transparency ratio associated with the i-th model, indicating its contribution to the final output. n denotes the number of approximate retrieved models.

Shadow guidance allows users to adjust the angle to observe the model before drawing the sketch. After the observation, the angle is fixed, and DualShape moves into the sketch drawing component. During the subsequent drawing process, the real-time retrieved model will also be displayed based on the previously determined observation angle. Fig. 13 shows how the background guidance changes with the sketch lines during the drawing process.

B. SKETCH OPERATION

We divided the model generation into parts retrieval and parts generation. The input sketch is also drawn in layers according to the 3D shape of the parts. After the current sketch is completed, users can create a new drawing layer. In order to distinguish between the different drawing layers, DualShape provides a dark (black) stroke to highlight the current layer. The sketch contents of the different layers can be fed to the back end of the part generator to obtain the corresponding model. In addition, DualShape provides a retrieval/generation method for users based on the distinction

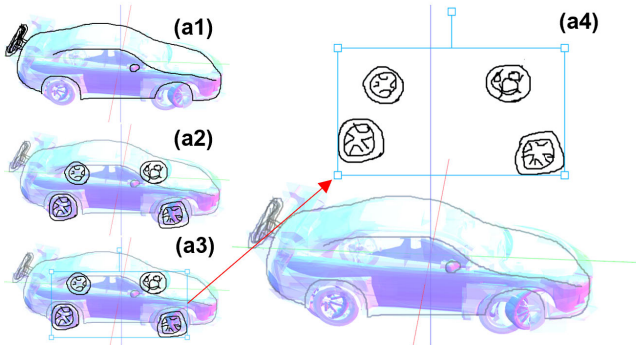


FIGURE 14. The state of the sketch area under different operations. (a1) the black stroke emphasizes the first layer currently being drawn - the car shell contour; (a2) the black stroke emphasizes the second layer currently being drawn - the tire contour; (a3) with the preview function, selecting a layer that needs to be edited again adds a selected state; (a4) a move operation is performed in the editable state, for example, the sketch of tires is moved from the position of (a3) to the position of (a4).

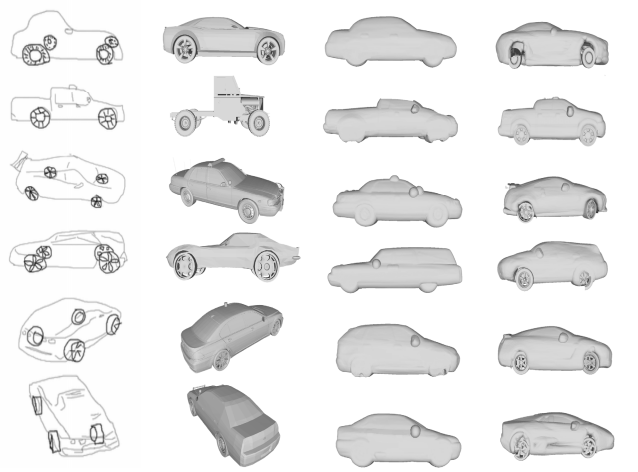


FIGURE 15. Comparison among different generation results. (a) the sketch inputs of the car shell and tires drawn by users, (b) the retrieved results by retrieval-only method corresponding to the user sketches, (c) the generated results from user-drawn sketches using generation-only method, and (d) the models generated by our proposed hybrid approach.

of the part features. Fig. 14 shows how DualShape interface changes when sketching different layers.

C. PREVIEW FUNCTION

DualShape provides a real-time preview function for sketches and parts, allowing users to keep track of the current progress and expected results of model generation. At the same time, users can click to select a preview layer, and the corresponding layer in the drawing area will be highlighted in the dark so that users can modify the current view. Users can also perform panning, zooming, and adding and deleting lines on the view.

D. MODEL ASSEMBLY

DualShape automatically performs the assembly of parts and displays the assembled result in real-time when users finish retrieval/generation tasks. If the user is unsatisfied with the current assembly, DualShape provides a mode for editing the assembled model. Users can select the parts they want to adjust in the edit mode. Models can be moved or scaled along X/Y/Z axis. Fig. 11 shows the translation and scaling of the assembly model.

V. RESULTS

A. IMPLEMENTATION DETAILS

We implemented a real-time drawing user interface with DualShape in Python on a Windows 10 platform, which is equipped with a 3.60 GHz Intel Xeon W-2223 CPU, GeForce RTX 3090 GPU. For the developed prototype, the average generation time of the shadow guidance was 0.82 seconds and the average execution time of the retrieval method was about 1.13 seconds. In contrast, the generation method had an average execution time of 4.23 seconds. This performance balance, where the system efficiently manages both quick retrieval and computationally demanding generation tasks, highlights the system's ability to handle diverse tasks, providing a stable user experience in both retrieval and generation activities.

B. DESIGN RESULTS

We invited sixteen participants to use the proposed system to design car models. Fig. 15 shows several examples of designed car models by users. By comparing the results with the same input sketches (Fig. 15(a)) in the retrieval-only and generation-only methods, it is verified that our proposed hybrid approach can preserve the structure of the connections between the model parts (the car shell and the car tyres), and also the richly-detailed features of the car tyres. Specifically, the user first sketched the car shell and used the generation method to generate different shells to satisfy the diversity of shells in car model design. For the tire part, the users simply drew the features of the tires and used the retrieval method to obtain the tire model from the dataset. After the initial automatic assembly according to the assembly rules (Section III-D), the user can manually adjust the positions and sizes of the parts to conform to the overall designs. It is verified that the entire design process was simple and user-friendly. The proposed system can reduce the complexity of model design, especially for novices without 3D modeling experience.

C. COMPARISON STUDY

To evaluate the effectiveness of DualShape, we conducted comparison studies with the state-of-the-art sketch-based modeling approaches: MeshSDF [31] and Sketch2Mesh [8]. MeshSDF takes the whole sketch as input and generates the model from a global perspective without focusing on the models' details. The results generated by MeshSDF tend to blur the details of the individual parts when they are

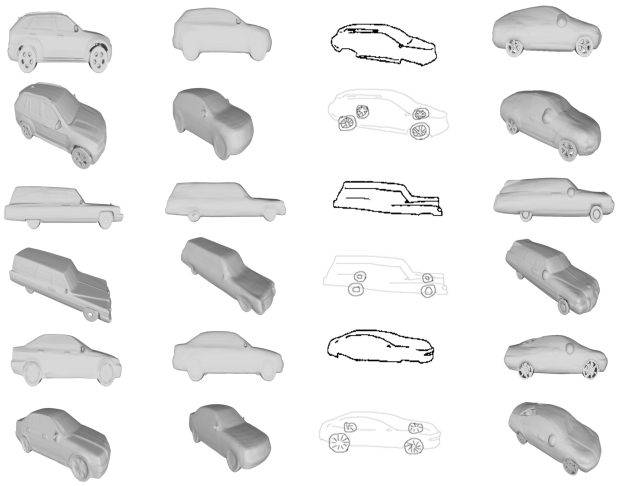


FIGURE 16. The ground truth models(a), the generated models (b) using MeshSDF with sketch inputs (c), and the model generated using our system with the same sketch inputs (d).

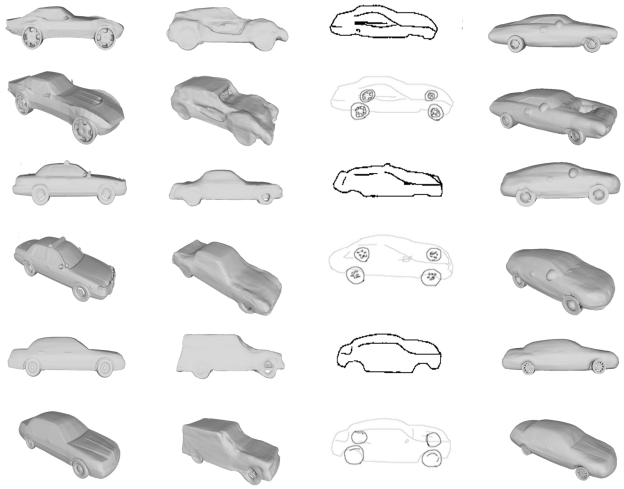


FIGURE 17. Figure (a) represents the ground truth models; (b) represents the model generated using Sketch2Mesh with the whole sketch in (c); (c) is the sketch input into our system, including the sketch of the car shell and the whole sketch with the tires added; and (d) is the model generated using our system with the sketch in (c) as input.

TABLE 1. Evaluation metrics of model quality.

Method	Metric	
	CD-$l_2 \cdot 10^3 \downarrow$	*NC* $\cdot 10^2 \uparrow$
MeshSDF	4.28	90.68
Sketch2Mesh	3.09	90.75
Ours	2.53	89.21

combined with each other. As shown in Fig. 16(b), since the car shells and the car tires are generated as complete units in MeshSDF, the connection structures between the car shells and the tires are not obvious. In addition, MeshSDF disregards the parts' local features. The car tires of each model almost do not have distinctive detail features.

The generated models of DualShape are shown in Fig. 16(d). In contrast to MeshSDF, DualShape considers the model generation as an assembly of two parts: the car shells and the car tires. This emphasizes the structural features when the parts are combined, such as the car shell having a distinct depression to fit the shape of the tires. These two parts are independent of each other but can be assembled into a complete model. Fig. 16(d) shows that the model generated by DualShape also emphasizes the detailed features of the tire parts. The tires have obviously distinguishable features.

Sketch2Mesh uses potential parameterization to denote and refine the 3D mesh to match its projection to the external contours traced within the sketch. However, this method ignores the fact that the model is an assembly of parts, neglects the structure of connections between components, and disregards the detailed features of the part model. The car models generated by Sketch2Mesh are shown in Fig. 17(b). We can observe that the generated model lacks local details, indicating that this method may only be capable of generating the general structure of the car model. The tires and the car shell are integral parts, rather than existing independently as parts in Sketch2Mesh. The feature structures of the connection between the car shell and the tires are blurred out. Meanwhile, the generated tire part has only a general structure; the stylized features inside the tire are not represented. The differences between the tires of different models are relatively minor. Fig. 17(d) shows the model generated by DualShape. Compared to the model generated by sketch2Mesh, the structures of the connection between the two parts of our car shell and tires are crisp. In addition, the tires with different styles of models are clearly distinguished from each other.

For the assessment of model quality, we utilized the 3D Chamfer Distance loss (CD-l_2) as a measure metric, where lower values indicate superior performance. The l_2 specifically refers to the Euclidean norm, which quantifies the dissimilarity by considering the squared distances between two point clouds. This metric is computed by sampling $N = 20000$ points on the reconstructed mesh to create the first point cloud $\mathbf{C}_1$, and N points on the ground truth mesh to create the second point cloud $\mathbf{C}_2$. The CD-l_2 is calculated as follows:

$$\text{CD-}l_2 = \frac{1}{N}\sum_{x\in\mathbf{C}_1}\min_{y\in\mathbf{C}_2}\|x-y\|^2 + \frac{1}{N}\sum_{y\in\mathbf{C}_2}\min_{x\in\mathbf{C}_1}\|y-x\|^2 \tag{7}$$

Additionally, we adopted a normal consistency measure (NC) metric, where higher values indicate better consistency. The normal consistency is defined as the average of the absolute dot product between normals in the reconstructed mesh G and their corresponding nearest normals

in the ground truth mesh R. The formula is defined as follows:

$$NC(G,R) = \frac{1}{|R|}\sum_{r\in R, g\in G}|r\cdot g| + \frac{1}{|G|}\sum_{r\in R, g\in G}|g\cdot r| \quad (8)$$

where G and R represent the normals in the reconstructed and ground truth meshes, respectively. $|R|$ and $|G|$ represent the number of normals in the ground truth and reconstructed meshes, respectively. This formula calculates the average absolute dot product between corresponding normals in the reconstructed and ground truth meshes, providing a measure metric of the consistency of normals between two meshes.

As shown in Table 1, our method performs better compared to MeshSDF (4.28) and Sketch2Mesh (3.09), with a value of 2.53 in $CD\text{-}l_2 \cdot 10^3$. The lower value of $CD\text{-}l_2$ indicates better alignment between the resulted model and the ground truth. Therefore, our method performs well in accurately capturing the geometry of the object, leading to a reduced Chamfer Distance. With respect to $NC \cdot 10^2$, our method achieves a value of 89.21, showing close competitiveness with MeshSDF (90.68) and Sketch2Mesh (90.75), which suggests that it maintains a high level of normal consistency.

VI. USER STUDY

We conducted a user study to evaluate the effectiveness of our user interface. Sixteen participants (including ten male and six female graduate students) were invited. They tested our system to generate car models and conducted a questionnaire after the experiment. The participants' evaluations of the user interface were analyzed statistically. Our questionnaire survey includes 3 aspects: overall evaluation, System Usability Scale (SUS) evaluation, and user interface specific function evaluation.

A. OVERALL EVALUATION

After the experiment, the participants are invited to rate, ''system functional integrity'', ''user interface convenience'', ''generated results satisfaction'', and ''conformity to expectations'' by using a 5-point Likert scale (1=strongly disagree, 5=strongly agree). As shown in Fig. 18, we analyzed the distribution of each indicator. The system's functional integrity is concentrated on rank 4, and the convenience of the user interface is in the range of ranks 3-4. The satisfaction with the results is also concentrated in rank 4, with most users agreeing that the generated results conformed to their expectations.

The mean value of users' rating of the ease of use of the interface is 3.88, and the standard deviation is 0.62. Most participants thought that our user interface was intuitive. The mean value of the participants' rating of the functional completeness of the UI is 4.06, with a standard deviation of 0.57. This shows that the functionality of our system is relatively complete. The satisfaction of the model obtained a mean of 3.94 and a standard deviation of 0.68. Most participants were satisfied with the model obtained.

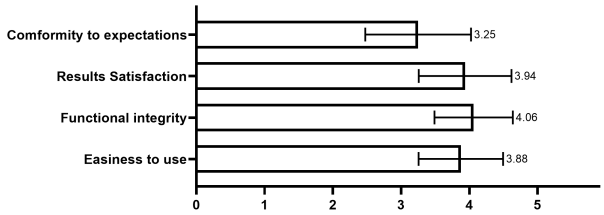


FIGURE 18. Overall evaluation results.

TABLE 2. Results of the post-experiment SUS metrics questionnaire. ⇑ indicates that higher scores are better; ⇓ for the other case. The total score is 81.67 out of 100.

	Questions	Mean	SD
1	I would like to use this system frequently. ⇑	3.75	0.68
2	I found this system unnecessarily complex. ⇓	1.44	0.51
3	This system was easy to use. ⇑	3.88	0.62
4	I would need the support of a technical person to be able to use this system. ⇓	1.75	0.45
5	I found the various functions in this system were well integrated. ⇑	4.06	0.57
6	I thought there was too much inconsistency in this system. ⇓	1.81	0.40
7	I would imagine that most people would learn to use this system very quickly. ⇑	4.19	0.40
8	I found this system very cumbersome to use. ⇓	1.31	0.48
9	I felt very confident in using this system. ⇑	3.88	0.50
10	I needed to learn a lot of things before I could get going with this system. ⇓	1.06	0.25

The mean value of ''How well the resulting model met expectations'' was 3.25, and the standard deviation was 0.77. Most participants agreed that the obtained model conformed to their expectations.

B. SUBJECTIVE EVALUATION

In addition to completing the overall evaluation, participants were invited to answer questions to evaluate DualShape based on the System Usability Scale (SUS).

Table. 2 shows the results of the participants' evaluations of the SUS metrics. All participants indicated that the overall interface is satisfying, they agreed that most users would be able to learn to use the system quickly. 75% of the participants commented that they would like to use the system frequently to design car models. 77.6% of the participants agreed that the interface functions were easy to use. In addition, 81.2% of the participants and 77.6% of the participants respectively supported that our system's functionality was adequately integrated and that they were confident while using the system. Within the SUS metrics, our system is scored 80.94 out of 100, which indicated that the overall usability of our user interface was excellent.

TABLE 3. Results of specific functions evaluation questionnaires.

	Questions	Mean	SD
1	Sketching in layers is in line with the drawing habit.	3.56	0.96
2	The shadow guide function can adequately assist me in sketching.	3.75	0.58
3	Shadow guidance is easy to use.	4.06	0.44
4	The generate function was useful for generating car shell models.	4.00	0.63
5	The car shells generated using the generation method meet expectations.	3.44	0.51
6	It was useful to use the search function to retrieve the models.	4.25	0.45
7	The model retrieved using the search function meets expectations.	4.50	0.52
8	The drawing function is easy to use	4.31	0.48
9	The delete function is easy to use.	2.38	0.50
10	The undo function is easy to use.	4.44	0.53
11	The download function is easy to use.	4.75	0.58
12	The preview function provided can be a good way for me to observe the model generation process.	4.12	0.72
13	The function of re-editing after selecting the preview layer is very useful.	4.06	0.44
14	The assembly function is useful.	4.31	0.48
15	The re-edit function is useful.	4.69	0.48
16	The re-edit function is easy to use.	3.63	0.50

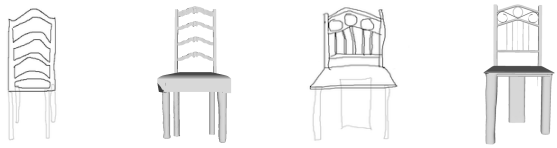
C. SPECIFIC FUNCTIONS EVALUATION

In order to determine the usefulness and convenience of the interface functions we designed, the participants were required to evaluate the specific functions they used. The evaluation is shown in Table 3.

We use a layer-by-layer part-sketching approach to generate parts, 71.2% of participants found this approach was consistent with their drawing habits, which allowed faster design. 75% of participants found that the 3D shadow guidance was helpful in quickly sketching the outline of the car, and 81.2% of participants indicated that our 3D shadow guidance function was intuitive while providing the closest reference model in real time.

In addition, 80% of the participants thought it was reasonable to use the generation method for car shells, and 68.8% of the participants thought the generated car shell models basically met their expectations. Specifically, 85% of the participants acknowledged the usefulness of the tire parts retrieval method, and 90% of them were satisfied with the retrieved models. Meanwhile, for the basic draw/delete/undo/download functions, respectively, 86.2%/47.6%/88.8%/95% of the participants thought they were user-friendly. For the preview and re-edit sketch functions, 82.4% of users thought the preview functions observed the model generation process well, 81.2% of participants acknowledged the necessity of the re-edit sketch function, but only 55% of users were satisfied with the re-edit function of the sketch.

For the ability to display the assembled model in real time, 86.2% of participants thought positively. 93.8% of the participants acknowledged the necessity and usefulness of the assembled model details adjustment, and 72.6% of the participants thought the function of model adjustment was user-friendly.



(a) Input sketch1 (b) Chair model1 (c) Input sketch2 (d) Chair model2

FIGURE 19. Examples of designed chair models. (a) and (c) represent the chair sketches drawn by the user as input. (b) and (d) are the results generated using the hybrid method for the user sketches (a) and (c), respectively.

VII. DISCUSSION

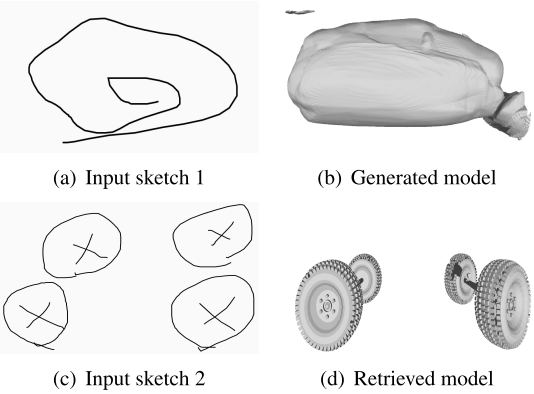
In this section, we discuss the feasibility of shape design with the proposed approach, limitations, and future work og this work. In the shape design section, we focus on the scalability of the proposed system and the diversity of designable models using our hybrid approach. In addition, we discuss the limitations of the proposed method, and give some failure cases to facilitate the observation of the limitations of the system. In response to these limitations, we provide possible solutions as future work.

A. SHAPE DESIGN

Our proposed framework can also be used in various shape designs, and its extensibility can be scaled to multiple model designs. We take chair design as a simple extension. The designed chair models by users are shown in Fig. 19. We partitioned the chair design into two parts: the body(back and seat) and the legs. The user can draw a simple feature sketch of the chair back and seat, and then retrieve the most similar part from the dataset. The user can draw the chair leg part and use the generation method to obtain the corresponding part. The obtained results indicate that the proposed system can be further extended to acquire various categories of models. The extensibility and adaptability of DualShape can be assured for the design of different models.

B. LIMITATIONS AND FUTURE WORK

One of the limitations of the proposed approach is to note that during the initial automatic assembly of models, parts of different kinds of models are required to be assembled according to their particular prior knowledge. In the Shape design section, when assembling the chair parts, we did not set rules for the initial assembly between parts, and the user need to manually adjusted the part positions and sizes, as shown in Fig. 19. In order to address this issue, we suggest adding a module that automatically learns the positional relationship and the the proportional relationship between shapes, which would then eliminate the need to set up different assembly rules for the shape design of each category object, and the automatic assembly between parts can be fully realised.



(a) Input sketch 1 (b) Generated model

(c) Input sketch 2 (d) Retrieved model

FIGURE 20. Part model generation/retrieval failure cases. (a) and (c) represent the sketches drawn by the user as input, (b) is the failure case generated by the sketch corresponding to (a) using the generation method, and (d) is the failure case retrieved by the sketch corresponding to (c) using the retrieval method.

Another limitation is that when the sketch is simple, the generated shell part model may have bad retrieved results. Some cases of generation and retrieval results are shown in Fig. 20. When the features provided by the sketch for retrieval are too sparse, the retrieval results may not satisfy the user's design intention. Therefore, it is a promising solution to add a contour optimization module to get sketches with richer structural features. Meanwhile, the sketch contour optimization module would also improve the model search results.

In addition, our system currently supports the construction of a specific category, (i.e., car models) that have limited suitability. We plan to expand the categories of generated artificial objects, such as airplanes, tables, and vases. The generated model categories can be improved with other complex structures, such as animals and human bodies. We also plan to expand the dataset used to provide more informative generation results. Finally, the models generated by our hybrid approach may differ in resolution, which can be solved by mesh optimization.

VIII. CONCLUSION

In this work, we propose a hybrid 3D part assembly-based shape design framework, DualShape, which applies the retrieval and generation method to obtain the corresponding part models, and then assembles them into a complete model. In addition, we adopted an assisted drawing method by using 3D models as shadow guidance. Based on the above framework and the assisted method, we have developed and implemented a user interface in practice. The proposed system can perform not only the generation tasks from sketches but also the retrieved part models to preserve the detailed model features. In addition to comparison experiments, we also conducted a user study to verify the effectiveness, usefulness, and convenience of the framework and interface.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] J. Delanoy, M. Aubry, P. Isola, A. A. Efros, and A. Bousseau, "3D sketching using multi-view deep volumetric prediction," *Proc. ACM Comput. Graph. Interact. Techn.*, vol. 1, no. 1, pp. 1–22, Jul. 2018.

[2] C. Ding and L. Liu, "A survey of sketch based modeling systems," *Frontiers Comput. Sci.*, vol. 10, no. 6, pp. 985–999, Dec. 2016.

[3] D. Du, H. Zhu, Y. Nie, X. Han, S. Cui, Y. Yu, and L. Liu, "Learning part generation and assembly for sketching man-made objects," *Comput. Graph. Forum*, vol. 40, no. 1, pp. 222–233, Feb. 2021.

[4] X. Du, Y. He, X. Yang, C.-M. Chang, and H. Xie, "Sketch-based 3D shape modeling from sparse point clouds," in *Proc. Int. Workshop Adv. Imag. Technol. (IWAIT)*, May 2022, pp. 714–719.

[5] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, "Sketch-based shape retrieval," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–10, Aug. 2012.

[6] G. Fahim, K. Amin, and S. Zarif, "Single-view 3D reconstruction: A survey of deep learning methods," *Comput. Graph.*, vol. 94, pp. 164–190, Feb. 2021.

[7] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, "A search engine for 3D models," *ACM Trans. Graph.*, vol. 22, no. 1, pp. 83–105, Jan. 2003.

[8] B. Guillard, E. Remelli, P. Yvernay, and P. Fua, "Sketch2Mesh: Reconstructing and editing 3D shapes from sketches," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13003–13012.

[9] X. Han, C. Gao, and Y. Yu, "DeepSketch2Face: A deep learning based sketching system for 3D face and caricature modeling," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Jul. 2017.

[10] Z. Huang, Y. Peng, T. Hibino, C. Zhao, H. Xie, T. Fukusato, and K. Miyata, "DualFace: Two-stage drawing guidance for freehand portrait sketching," *Comput. Vis. Media*, vol. 8, no. 1, pp. 63–77, Mar. 2022.

[11] Z. Huang, H. Xie, T. Fukusato, and K. Miyata, "AniFaceDrawing: Anime portrait exploration during your sketching," in *Proc. ACM SIGGRAPH Conf.*, New York, NY, USA, Jul. 2023, pp. 1–11.

[12] E. Iarussi, A. Bousseau, and T. Tsandilas, "The drawing assistant: Automated drawing guidance and feedback from photographs," in *Proc. 26th Annu. ACM Symp. User Interface Softw. Technol.*, New York, NY, USA, Oct. 2013, pp. 183–192.

[13] T. Igarashi and J. F. Hughes, "A suggestive interface for 3D drawing," in *Proc. ACM SIGGRAPH Courses*, Aug. 2007, pp. 173–181.

[14] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: A sketching interface for 3D freeform design," in *Proc. ACM SIGGRAPH Courses*, Aug. 2007, pp. 409–416.

[15] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3779–3788.

[16] Y. J. Lee, C. L. Zitnick, and M. F. Cohen, "ShadowDraw: Real-time user guidance for freehand drawing," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–10, Jul. 2011.

[17] C. Li, H. Pan, Y. Liu, X. Tong, A. Sheffer, and W. Wang, "Robust flow-guided neural prediction for sketch-based freeform surface modeling," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–12, Dec. 2018.

[18] J. Li, C. Niu, and K. Xu, "Learning part generation and assembly for structure-aware shape synthesis," 2019, *arXiv:1906.06693*.

[19] A. Limpaecher, N. Feltman, A. Treuille, and M. Cohen, "Real-time drawing assistance through crowdsourcing," in *Proc. AAAI Conf. Hum. Comput. Crowdsourcing*, vol. 1, Nov. 2013, pp. 101–102.

[20] J. Loffler, "Content-based retrieval of 3D models in distributed web databases by visual shape information," in *Proc. IEEE Conf. Inf. Vis. Int. Conf. Comput. Visualizat. Graph.*, Jul. 2000, pp. 82–87.

[21] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, Aug. 1987.

[22] Z. Lun, M. Gadelha, E. Kalogerakis, S. Maji, and R. Wang, "3D shape reconstruction from sketches via multi-view convolutional networks," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 67–77.

[23] C. Ma, X. Yang, C. Zhang, X. Ruan, and M.-H. Yang, "Sketch retrieval via local dense stroke features," *Image Vis. Comput.*, vol. 46, pp. 64–73, Feb. 2016.

[24] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4460–4470.

[25] K. Mo, P. Guerrero, L. Yi, H. Su, P. Wonka, N. Mitra, and L. J. Guibas, "StructureNet: Hierarchical graph networks for 3D shape generation," 2019, *arXiv:1908.00575*.
[26] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 909–918.
[27] G. Nishida, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and A. Bousseau, "Interactive sketching of urban procedural models," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–11, Jul. 2016.
[28] C. Niu, J. Li, and K. Xu, "Im2Struct: Recovering 3D shape structure from a single RGB image," 2018, *arXiv:1804.05469*.
[29] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 165–174.
[30] Y. Peng, Z. Huang, C. Zhao, H. Xie, T. Fukusato, and K. Miyata, "Sketch-based human motion retrieval via shadow guidance," in *Proc. Nicograph Int. (NicoInt)*, Jul. 2021, pp. 42–45.
[31] E. Remelli, A. Lukoianov, S. Richter, B. Guillard, T. Bagautdinov, P. Baque, and P. Fua, "MeshSDF: Differentiable iso-surface extraction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 22468–22478.
[32] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, vol. 3. Cambridge, U.K.: Cambridge Univ. Press, 1999.
[33] D. Zhang. (2017). *Opensse: Open Sketch Search Engine*. [Online]. Available: https://github.com/zddhub/opensse

XUSHENG DU received the B.E. degree from the School of Physics and Electronic Engineering, Guangzhou University. He is currently pursuing the Ph.D. degree with the Japan Advanced Institute of Science and Technology (JAIST). His main research interests include computer graphics, 3-D models, machine learning, and interactive user interface.

TIANYU ZHANG received the B.E. degree from the School of Computer Science and Technology, Xidian University, in 2019. He is currently pursuing the Ph.D. degree with the Japan Advanced Institute of Science and Technology (JAIST). His main research interests include computer graphics, machine learning, and interactive user interface.

HAORAN XIE (Member, IEEE) received the Ph.D. degree from the Japan Advanced Institute of Science and Technology (JAIST), in 2015. After that, he was with the Computer Science Department, The University of Tokyo, from 2015 to 2018; and an Assistant Professor with JAIST, from 2018 to 2020. He has been an Associate Professor with JAIST, since 2023. His main research interests include interactive computer graphics and user interface. He is a member of ACM, IPSJ, and IIEEJ.