www.cadanda.com
Computer-Aided Design
AND APPLICATIONS

Sample Management System Based on Functionality Through User-Defined Geometric Constraints

Kemal M. Dogan[1] , Hiromasa Suzuki[2]

[1]Yildiz Technical University, kmdogan@yildiz.edu.tr
[2]The University of Tokyo, suzuki@den.t.u-tokyo.ac.jp

Corresponding author: Kemal M. Dogan, kmdogan@yildiz.edu.tr

Abstract. Designers often constrain their conceptual design as they sketch the idea to make sure the final concept solution has the expected functionality. Proposed system enables user-defined constraints created as mathematical formulas according to a syntax provided using geometrical features, ready-to-use functions, and mathematical operators. In this way, the users will be able to use computers' generative powers for their specific needs without being limited by built-in constraints. Defined constraints can be used in a sampling method as objectives to explore new design samples with better performance or filter out the undesired designs after the samples are generated. Thanks to its flexibility that lets the users define their constraints, the proposed method will also enhance the usability of generative design studies.

Keywords: computer-aided design, human-computer interaction, conceptual design, constraint development, product design
DOI: https://doi.org/10.14733/cadaps.2023.190-212

1 INTRODUCTION

A product design is an iterative process that often starts from the conceptual stage, where design problems defined based on customer needs are solved by designers [33]. In fact, the outcome of the conceptual stage is significant for the success of the overall design progress since insufficient solutions cannot proceed further in the follow-up procedures. Although those unfitted solutions to the product specific design problems are eventually discarded, time is wasted until then. The aforementioned problems are defined by considering various conditions such as the expected performance of the product, functionality, cost, and how customers feel about it. In addition, as the variety of most of the products has increased, aesthetics become a critical criterion to highlight a product among all other competitors.

Designers often create and test various design solutions to explore a product design that satisfies engineering and industrial design requirements, such as functionality and aesthetics [20]. Because of such diversity in the requirements, the designers are forced to be creative in a limited area in the design space of an infinite number of solutions. The space exploration quality is important in finding target design; but the designers' imagination, their experience, and the time they have before the final decision cause additional limitations. Outsourcing computer-aided design (CAD) methods can increase the efficiency of such explorations of the designs [27]. These methods can auto-generate many concept designs according to prerequisites set by the designers to shorten the time need for solution generations, thus improve the quality of the design process.

Generative design approaches are incorporated with pre-determined objectives and constraints so that they can suggest only potential design solutions among all generated design samples during the exploration process. Such a procedure of exploring the designs is acknowledged as sampling. Even though there are several suggested sampling methods that use geometric constraints to get feasible designs [14], they offer limited options by being useful only within the context of their study. Because of this, users cannot easily extend the proposed approaches when needed; thus, those studies suffer from being inflexible to be adapted for unique problems. This study proposes an assisting method for eliminating undesired design solutions programmatically, letting users to define their geometric constraints that provide the desired flexibility in terms of adaptability for special design needs.

Our approach adopts an earlier generative design method developed by [10], where profile curves of existing designs are modified to produce new samples. The generated samples are constrained with some rules defined using geometrical features of simple shapes, like circles and triangles, to avoid unacceptable solutions are being produced. In their study, the geometrical shapes are defined using control points of parametric curves and the rules defined using geometrical entities. For example, a circle that passes through two endpoints of a curve segment is created and the other points that create the control polygon must be on only one side of the diameter. This constraint is used when inflection in a curve segment cannot be accepted. As another example, edges of triangles, which are built over the control points, may not be overlapped in order to prevent having cusps and loops on the relevant segment. There are also similarity-based constraints that are anisotropy [4] and Modified Hausdorff distance [12] that apply to establish diversity between the generated designs and yet there is a desired similarity to the original design to maintain existing functionality. Although such constraints can prevent the unacceptable failures in the design, they cannot be altered or related to limited features that might affect the performance of a design. In this study, we propose a constraint development technique and embed it into the generative system suggested by [10] to make it flexible for its users; who are designers or people without experience in design want to check basic functional requirements that can be formulated with proposed mathematical functions. In the suggested system, geometrical entities of the designs, such as points on profile curves and curves used to construct cross-sections, are used to create constraining equations. In this way, besides measuring the specific shape features such as width, height or volume of a design; one can also consider some design ratios similar to proposal of [38] made for yacht design performance evaluation.

In addition, the suggested system by [10] allows only a single profile curve to be extruded or revolved to create the intended design. In order to show the capability of our constraint system over free-form shapes, we extend their approach using two or more profile curves together to form a design. The additional profile curves are used as cross-sections and guide curves to create swept or network surface models. More information on this matter will be given later in this paper.

Design process for the proposed system

Two different design processes may be used to integrate the proposed system depending on how the constraints are used:

- **Filtering designs after all samples are generated:** Figure 1 displays a conceptual design process in which new design samples are derived from an initial design first, geometrical constraints are defined and then applied to filter out the undesired ones. Since the filtration does not wipe off the designs, the constraints can be redefined and applied at any time needed.

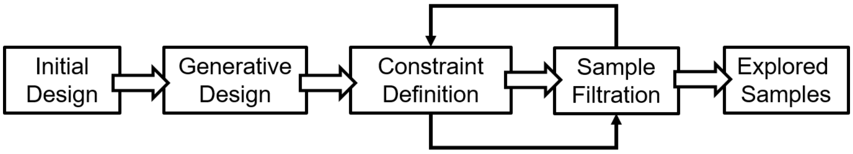


Figure 1: A suggested conceptual design process, where the geometric constraints are applied after all the samples are generated by a generative design method.

- **Integrating the geometric constraints to the exploration algorithm:** Figure 2 shows a design process in which constraints are defined and integrated into a generative design method before the sampling. This usage evaluates the sample immediately after it is generated and discards it if it does not satisfy the constraints. This option is recommended when constraint evaluation takes time, as the user does not have to change the constraint anymore and need to see only target design solutions at the end of the progress. Changing the constraints requires re-sampling.

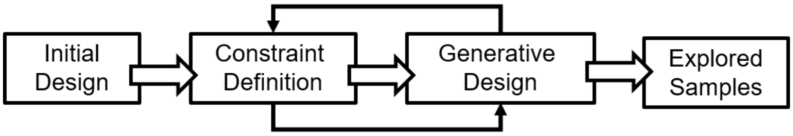


Figure 2: A suggested conceptual design process where the geometric constraints are defined before generating the samples and the samples are generated satisfying such constraints.

Overview of the proposed approach

In this study, we suggest an approach in which a user can formulate geometrical constraints specific to a product's functionality to narrow down the solutions. To do this, a sample management interface was offered in which a formula can be created using similarity measures and geometric constraints defined by users using functions such as distances between points, width, height and volume of the shape. The sample management starts with clustering the samples using the k-means method and each cluster is then organized individually according to the feature distance calculated by user defined formula. In this way, similar designs are grouped in same clusters and lets a user compare them based on desired feature defined as formula, which is also defined by the user. Users can set a range with desired maximum and minimum values to filter out the samples whose formula results are out of the range. The samples can also be analyzed with a graph of one formula output versus another. Besides, the interface comprises several areas with various purposes, such as generating new samples from a selected sample, editing the sample, exporting samples to be converted into three dimensional (3D) models in a CAD tool, and pair-wise comparison of the selected samples based on a user-defined formula. We believe that this study will encourage the designers to use computers more in the early stages because of the flexibility of the system that provides users with an easier conversion of a 2D sketched idea to a 3D model besides the power of design generation capability.

In this paper, Section 2 explains the related works and compares them with our approach. The generative design method for generating the samples and surface modelling techniques to represent them is described in Section 3. The sample management software and constraint development method that allows user-defined geometric constraints, including results, is elaborated in Section 4. Section 5 and Section 6 make brief discussions, gives the conclusions and suggests possible future directions.

2 Related Works

2.1 Sketching for Conceptual Design Stage

Many studies in the literature are focused on topics such as assessing whether such sketching is necessary [3], the relationship between quality sketching and creativity [24], and designers' behaviour while sketching to understand their way of thinking during the design process and to aid them by building up assisting approaches[11, 36]. A product design is an outcome of a design process that covers several stages, including establishing a need-phase on the basis of market demands, defining design problems, and exploring solutions for the defined design problems [40]. The solutions are usually formed as 2D perspective sketches in the conceptual design stage, and the promising solutions are converted into orthographic views to give dimensional details before delivering them to the later stages, where 3D CAD models and prototypes are created to test the functionality of the resulting products. Despite recent technological advances, designers often sketch their ideas to use time efficiently to generate more solutions [2]. Our study focuses on testing fundamental functionality of 2D sketched ideas, hence improving assessing conditions before moving on to later stages of the product design.

2.2 Human-Computer Interaction (HCI)

Despite the general aim of such studies being to find optimal solutions through computers, human interaction to guide the underlying systems is still needed to fill the gaps that occur due to the remaining limitations of the technology. For example, suitable HCI systems can monitor and learn customers' aesthetic feelings through surveys with user-friendly interfaces [8] or even biosensors [9, 29]. As a result, the idea generation process can achieve its objectives efficiently. Some studies have claimed that traditional CAD tools do not support designers' intuition, and suggested new methods such as haptic interfaces for virtual environments [35] and digital sculpting [1] to overcome such drawbacks. There are also other HCI systems with touchscreens, pen input and haptic interface that have been developed to support designers in the development of conceptual designs, including the sketching phase [42, 30]. However, a quality HCI system gives users a sufficient level of room for process control while saving time and energy, minimizing user interaction to provide fast and reliable outcomes [5]. The proposed design interface in this study lets users define their constraints with basic interaction to let them have full control over the constraints, thus on the sampling procedures.

2.3 Design for Functionality and Aesthetics

A design often needs to satisfy requirements of both aesthetic form and functionality to be competitive in the market [41]. Functionality is usually given priority in the design, with aesthetics built around such engineering design, providing the design with a "gift wrap", as [40] describe. However, there is also a collaborative way in which industrial and engineering designers work together on the form that directly influences both the function and the aesthetics of the geometry from the beginning of the conceptual design stage. Function block diagrams and their extensions that describe the form and function parameters have been widely studied to accomplish such task [39]. Some other studies have focused on the abstraction process to obtain functional properties related to design and explore the essential functional solutions accordingly [7]. Our study is dedicated to providing flexibility in defining functional constraints based on the geometric form of the design. Unlike the other laborious studies that require problem specific initial treatments that are difficult to change, our system provides an interface where functions are represented by simple formulas, and built-in functions are available to shorten the formula and allows users to make quick interactive modifications.

2.4 Generative Design and Constraining

Generative design methods theoretically can produce an infinite number of novel and unbiased ideas; thus, they need constraints and objectives to guide the process of space exploration [29]. There are various generative design methods proposed in the literature [37]. Shape grammar is one method, in which various basic shapes are pre-defined and brought together according to some rules to create new product designs [34, 32]. Some studies use a parametric approach, where design parameter values are bounded and constrained to generate a finite number of feasible design solutions [22, 18]. Genetic algorithms are also commonly used as a generative method, which cross-over and mutate shape features of initially given designs to find new ones that satisfy pre-defined design objectives [17, 6]. The optimization-based approach operates through topological modifications and explores the designs based on visual and physical properties such as weight and stress [21, 16, 26]. Although important performance criteria are constrained by the system being allowed to generate or navigate between the desired design samples, these studies nevertheless still often have limited options in terms of constraints. Our system will allow users to define constraints they need based on the geometrical entities of the profile curves.

3 Sample Generation with a Generative Design Method

The user-defined geometric constraints are specific to designs and applied to filter out the ones with undesired functionality. The designs that will be examined in this study can be generated with any generative design technique, if characteristic curves that define the shape profile can be extracted out of the generated design samples. As a pilot implementation of our sample management approach, we create the designs in this study as surface models from one or more profile curves using the surface methods elaborated under the next subsection. To create the profile curves, an example-based generative system proposed by [10] has been adopted. In this system, a user is expected to supply control points to create the profile curves as composite Bézier curves. The profile curves are then automatically represented using some geometrical shapes, such as circles and triangles. The features of these geometrical entities are later used as constraints to ensure the obtained designs are feasible. The similarities between the triangles are also used to control the similarity and diversity among the generated samples. Note that mathematical representations of such feasibility and similarity constraints are beyond the scope of this paper and will not be detailed further. On the other hand, their system supports only a single profile curve to work with product features seen only from one viewpoint. To utilize some functional constraints that require geometrical features in 3D space, the system has been extended by combining multiple curves obtained from different viewpoints. A design software to define such multi-profiling is given in Fig. 3. Such extension allows defining more complicated and 3D-like functional constraints over a sketch sample, as well as allowing the user to generate various free-form objects, to enable multiple viewpoints to be studied together.

3.1 Composite Bézier Curve Definition

The composite Bézier is defined as joining Bézier curve segments to be one point is common at the ends of the relevant segments. A composite Bézier curve D can be represented by the following parametric equations:

$$D = \left\{ s^i(t) | i = 0, ..., K-1 \right\}, \qquad \begin{array}{l} s^i(t) = \sum_{j=0}^{n} B_j^i(t) P_j^i \\ 0 \leq t \leq 1 \\ s^i(1) = s^{i+1}(0) \end{array} \tag{1}$$

where K is the number of the segments, B is a blending function, P refers to the control points at each curve segment, n is the degree of the relevant segment and t is its parameter.

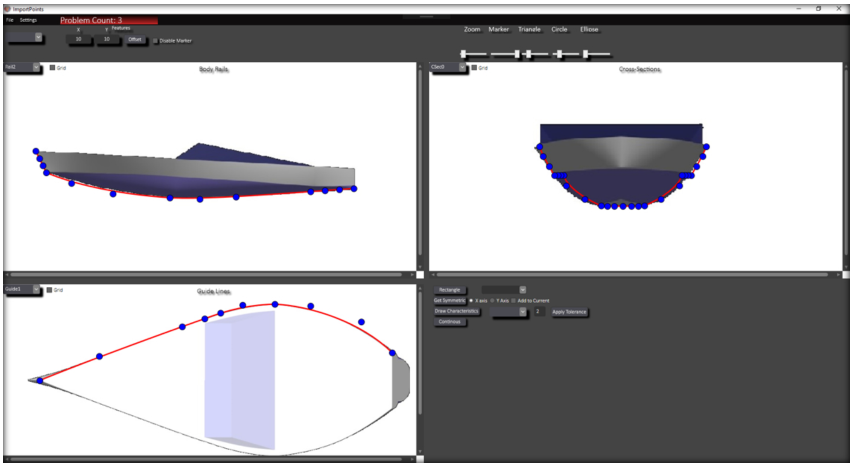


Figure 3: The interface for adding points to the multiple profiles of different views, in which the left upper represents the side view of the design example that is used to define the top and bottom guide curves; left bottom design is the top view used to define left and right guide curves, and the right upper view is the front view to define the cross-sections.

3.2 Surface Methods

The generated profile curves create the surface model of a design by extrusion or revolution when only a single profile curve is given as the input, and swept or network surface models are created when multi-profile curves are available. With swept or network surface models, the multi-profile curves are categorized as guide curves and cross-sections. Although up to four guide curves can be defined, the maximum number of cross-sections equals the number of endpoints of the curve segments, since each cross-section must be linked to an endpoint of a segment.

From now on, a single profile curve will be denoted as κ, while κ^g refers to a guide curve and κ^c represents a cross-section (Fig. 4). Thus, a surface model is represented by ζ that shows a list of such curves used by a relevant surface method.

- **Extrusion:** This method sweeps a single closed profile curve κ_0 in a direction of its normal vector r with magnitude of extrusion length z to create an extruded surface model $\zeta^e = \{\kappa_0\}$ (see Fig. 4a).
- **Revolution:** The surface model $\zeta^r = \{\kappa_0\}$ is obtained by rotating a single open curve κ_0 around an axis $\overleftrightarrow{L_1}$ that passes through the rightmost or leftmost endpoint of the curve to be $\overleftrightarrow{L_1} \parallel \overleftrightarrow{L_2}$ where $\overleftrightarrow{L_2}$ is defined as $x = 0$ (Fig. 4b). If the shape is rotated around an axis to be $\overleftrightarrow{L_2}$ is a line with the equation of $y = 0$, then $\overleftrightarrow{L_1}$ must pass through the uppermost or bottom-most endpoint of the curve to be revolved.
- **Sweeping:** A swept surface model can be defined with up to two guide curves and user defined number of cross-sections $\zeta^s = \{\kappa_0^g, \kappa_1^g, \kappa_0^c, \kappa_1^c ...., \kappa_n^c\}$, where κ_0^g and κ_1^g are the guide curves to sweep the cross-sections (Fig. 4c). In case of symmetry, κ_1^g is not defined but obtained by mirroring κ_0^g. Each cross-section κ_i^c is defined with index of control point that will be coincident for positioning, and κ_i^c can be repeated by amount of user input. Each cross-section is fitted between guide curves by scaling with scaling values S_x and S_y for x and y directions along κ_0^c until where κ_{i+1}^c starts.
- **Network Surface:** The network surface fits a surface through a curve network created by the intersections of three or four profile curves and a user-defined number of cross-sections (Fig. 4d); thus a

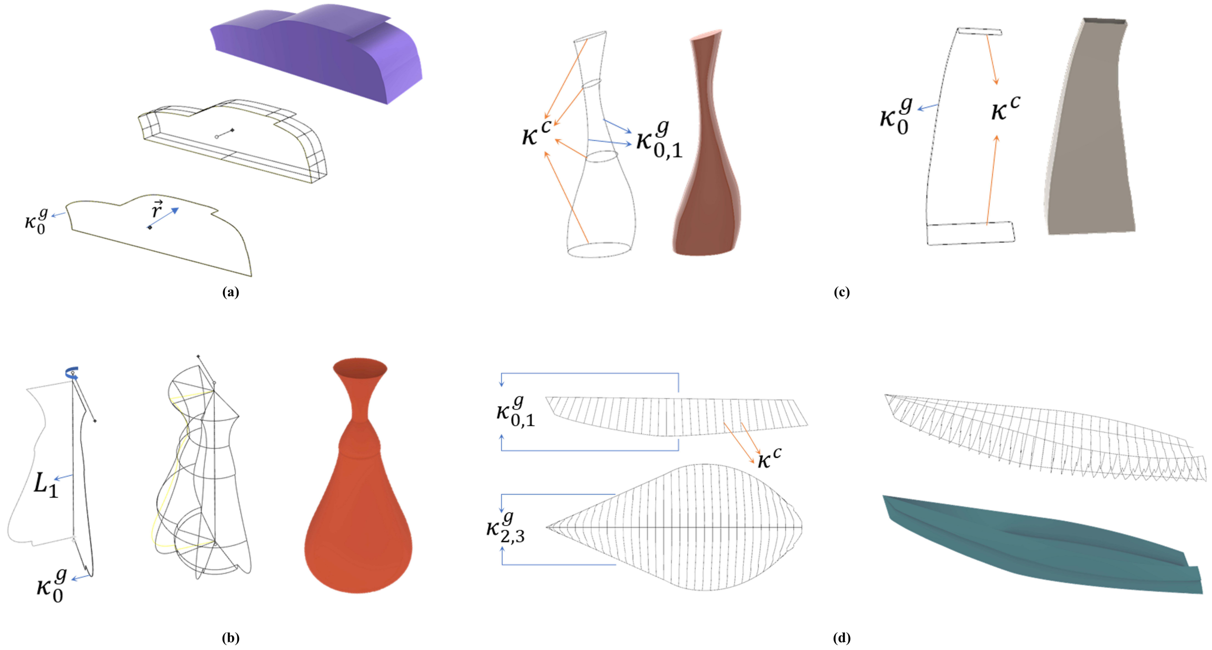


Figure 4: (a) Extrusion surface obtained by sweeping a 2D closed curve along a vector that is normal to its plane. (b) The surface is obtained by rotating an open 2D curve about an axis. (c) The surface obtained by cross-sections and two guide curves on the left, one guide curve on the right. (d) Network surface is created using multiple cross-sections four guide curves.

network surface is $\zeta^n = \{\kappa_0^g, \kappa_1^g, \kappa_2^g, \kappa_3^g, \kappa_0^c, \kappa_1^c...., \kappa_n^c\}$. In the design system, the cross-sections κ_i^c can be defined in any number user needs, but we recommend setting them for the locations only where the shape changes. The component that is used to create a 3D model also can derive a user-specified number of cross-sections to be placed with equal intervals. To do this, each of the profile curves into equal pieces and a cross-section is copied and affine transformed to be fitted between the corresponding endpoints of the pieces.

Figure 5 illustrates some example samples created by the sweep and network surface methods.

4 Sample Management

4.1 Sample Management Interface

Sample management interface (see Fig. 6) manages many samples by analyzing and filtering them using a formula created utilizing similarity metrics and user-defined geometric constraints. Such similarity metrics are anisotropy ratio [4] and modified Hausdorff distance (MHD) [12], which are calculated against the sample at the center of the interface using control points and the triangles that were mentioned earlier. In this section, we will focus on the geometric constraints that are related to product functionality, and we again refer to the study of [10] for more information about the similarity metrics. The geometric constraints are defined as mathematical functions and how to define them is explained in the next section.

Before any constraint applies to filter out the samples, k-means clustering [15] is employed to divide the samples into a user-specified number of clusters to make the management of them easier. In this way, the

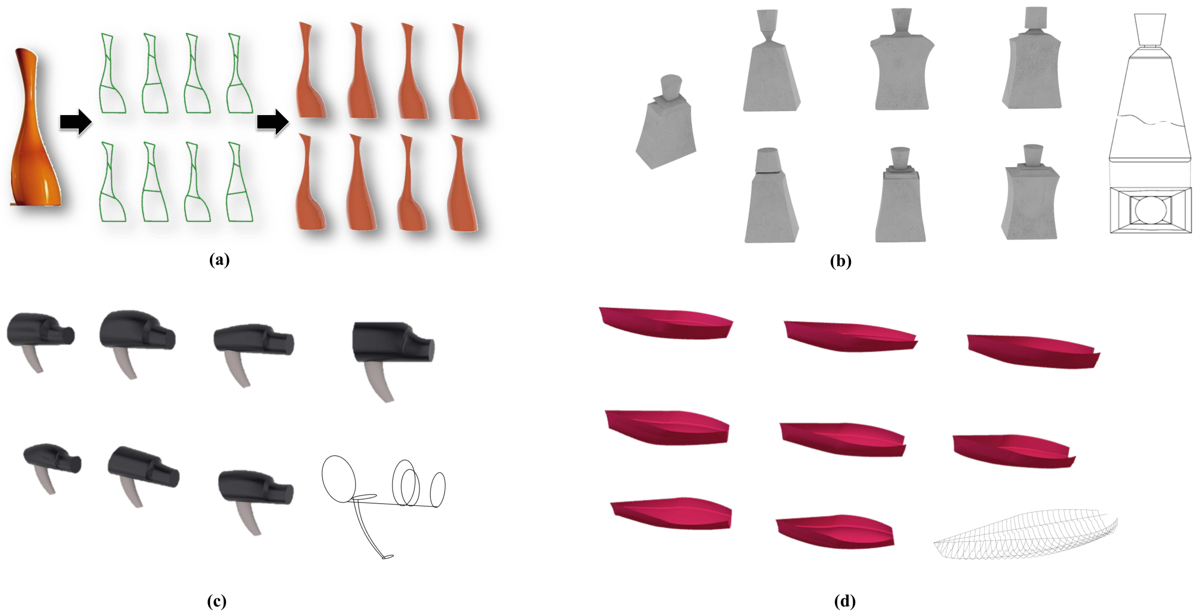


Figure 5: (a) Vase samples (on the right) created from the images (on the left) with sweep method using two profile curves and a single closed cross-section. (b) Bottle samples created with the sweep method using two profile curves and multiple closed cross-sections. (c) Hairdryer samples generated by sweeping a single closed cross-section for its upper body. (d) Hull samples generated by network surface method using four profile curves and a single open cross-section.

users can group similar designs and shape features that are formulated could be examined further within a relevant group. The very first samples shown are the determined centroids and the exemplar profile $\kappa_{initial}$ that the user-provided. In Figure 7, the clusters' centroids are placed surrounding the $\kappa_{initial}$ (in red). When a centroid representative sample is selected, the centroid goes to the center and becomes $\kappa_{initial}$ at that moment and its cluster members are located around it regarding shape feature distances calculated by the user-defined formulas. The surrounding samples are colored between light and dark blue, showing the highest and lowest value computed by a selected formula. Figure 8 shows some options about possible sample organization. Mainly, the samples can be at a distance to create a circular shape around the center or to be the ones with low values computed by a formula closer to the center sample. In addition, the samples can be organized in a way that the computed formula values are sorted to order the samples getting increased in the clockwise or counterclockwise direction.

The modules of the proposed interface are explained below with corresponding regions named by capital letters in Fig. 6:

- **A:** Main canvas is where the samples are displayed. The organization and color of the samples are determined based on similarity or a user-defined formula.
- **B:** Any sample can be dragged from "A" to "B" to be stored in a list.
- **C:** The samples in "B" are pair-wise compared based on a formula.
- **D:** The formulas are defined with *Formula Interface* (Fig. 10) and selected to filter out the samples using sliders. The sliders define upper and lower values such that if the formula value of a sample is

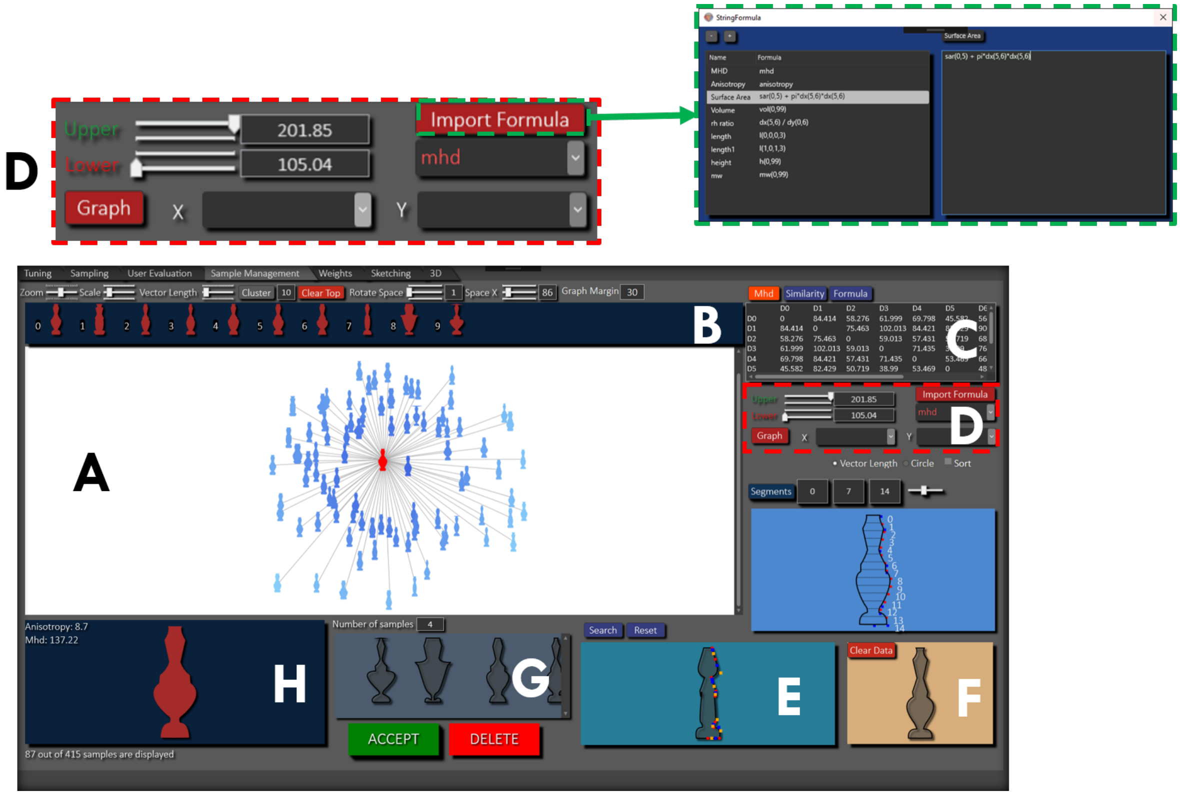


Figure 6: Sample management interface that filters and analyzes the samples based on similarities or user-defined formulas.

out of the range; the sample is removed from the canvas temporarily. A graph can be created over "A" using two formulas for X and Y axes (see Fig. 9).

- **E:** Any sample can be dragged here to manipulate its points. The modified sample can be dragged from "E" to any other region, so it can derive new samples in "G", can be exported to Rhino [28] in "F", or can be compared with others in "B". In addition, there is a search button that displays the samples in "A" and organizes them based on their similarity to this modified sample.
- **F:** The samples dragged here is exported to a file shared with Rhino so that its 3D model is simultaneously created.
- **G:** Any sample dragged into this region is used as an exemplar profile and a single chain algorithm is executed to create new samples from this sample. Through the "Accept" and "Delete" buttons, the generated samples can be added to the original list and or it can be erased.
- **H:** When the mouse cursor enters inside the boundary of a sample in any other region, its large-scaled version is displayed here.

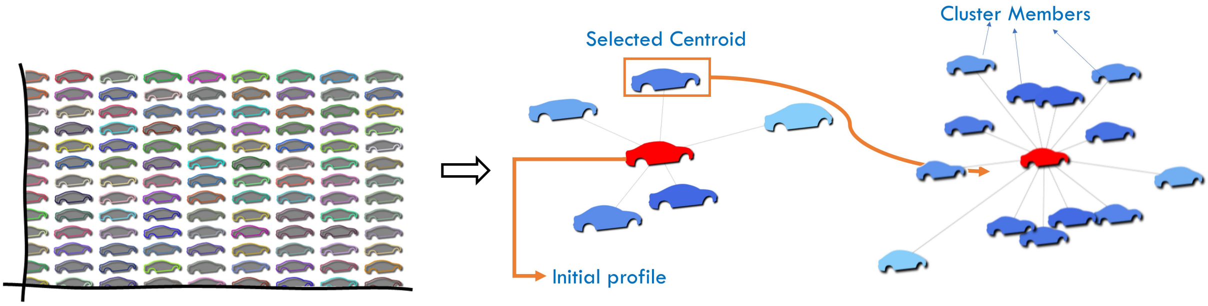


Figure 7: Illustration of generated samples (on the left) are clustered into five groups and represented by the clusters' centroids (in the middle). When a centroid is selected, relevant cluster members are displayed around the centroid to the user (on the right).

4.2 Formula Definition

A user builds the formulas using the profile similarities calculated between the corresponding curves as *anisotropy* and *MHD* metrics, or defining the geometrical constraints for the samples. The proposed interface lets the user define the geometric constraints using functions presented in Table 1. The functions' variables are the labels, which are automatically assigned for the points on the curve with user-specified equal intervals of the profile curve. In addition, if the user defines a cross-section for sweeping along the profile curves, the cross-sections are at even intervals and labelled to be called by a function. Figure 10 shows the interface of formula definition for a sample, where the points and the cross-sections are labelled. dx (Delta X) is a function that calculates horizontal distance between two points. For example, in Fig. 11, $dx(s23, s13)$ is the distance between points labelled as $s23$ and $s13$. Then, the samples are filtered and organised based on these distances of the samples as illustrated in Fig. 11 to be the distances are getting increased in a clockwise direction. In a same way, dy (Delta Y) compute vertical distance, l (length) computes Euclidean distance between two points, and w (Width) finds maximum dx and h (Height) find maximum dy of the shape. Besides, these functions can be incorporated using the basic mathematical operators that are addition, subtraction, multiplication and division. Note that there are also trigonometric functions to be used to create the geometrical constraints. Function ud is used to call any previously defined formula by its declared name within a new formula. This provides reuse for the created formulas and lets the user define nested, thus shorten formulas. Figure 12 shows simple geometric constraints applied on generated car samples and two extreme models obtained according to the outcomes of relevant formulas.

Interpreting the formula and determining the priority of the operations require a proper syntax. To do the job, the Reverse Polish Notation method [23] has been adopted, and the algorithm provided by [31] has been modified and used for the implementation.

Note that the formula definition interface let users define and store the constraints into different configurations. In this way a user can tryout different confliction free constraint variations in separated configurations.

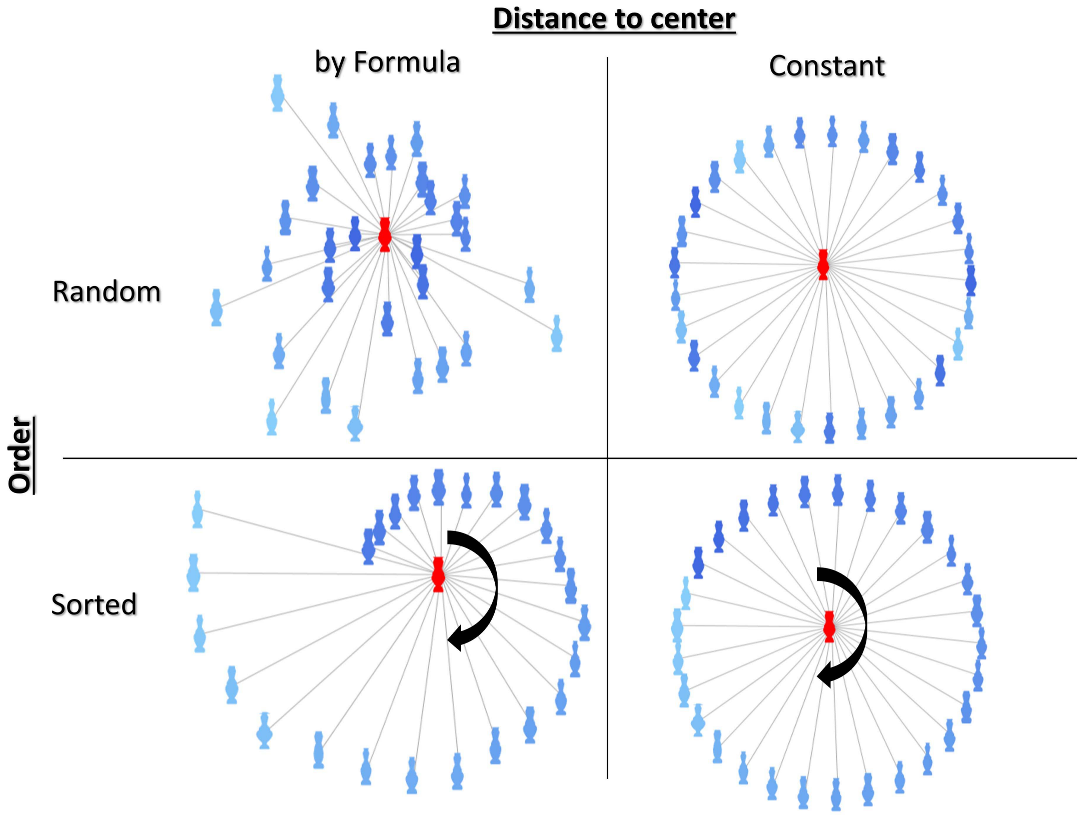


Figure 8: Sample organization options based on the distance to the center and the order. Here, the distance holds a value that is calculated by the user-defined formula or a constant value of user selection. The order of the samples in the frame could be random or sorted to be from the smallest value calculated by the formula to largest.

Table 1: Function list to define formulas.

Name	Functions	Name	Functions
Math Operators	+,-,*,/	Length	l
MHD	mhd	Angle	an
Anisotropy	sim	Delta X	dx
User Defined	ud	Delta Y	dy
Volume	vol	Point X	px
Trigonometric Functions	tan, sin, cos, atan, asin, acos, tanh, sinh, cosh	Point Y	py
Area	ar	Surface Area	sar
Width	w	Height	h
Min. Width	mw	Min. Height	mh

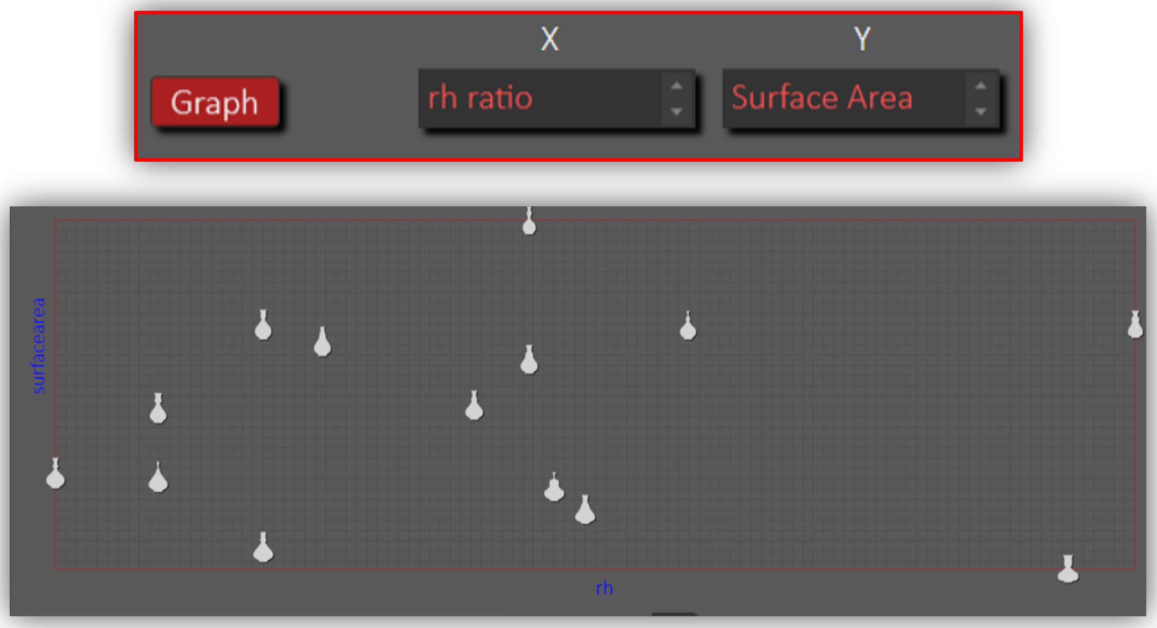


Figure 9: A graph of r/h ratio versus *surface area* is defined in region **D** and displayed in region **A** of the sample management interface.

Area:

One of the functions is *ar* used for computing area (see Fig. 13a)) approximately using the following Shoelace formula [25] such that the area is closed by a polygon created by the points on the profile curve (see Fig. 13b):

$$A = \frac{1}{2}|\sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n|$$

Volume:

Volume may not be used for creativity; however a generative design technique can generate many creative design samples and volume can be used as constraint to check on the weight of the product, assess its sustainability or simply the capacity. Volume is the amount of space occupied by a 3D object. However, as the generated designs are in 2D form, the volume function has been proposed to estimate the approximated volume using the two-dimensions of the shape. In this way, the need of 3D conversion of the generated design will be eliminated, and the time spent for the volume calculation will be saved. To compute the volume of a sample, pyramidal frustums are defined between consecutive intervals where the top and bottom areas indicate the areas of relevant cross-sections (see Fig. 14 and Fig. 15). The shape volume is then computed as the sum of all volumes of these pyramidal frustums. Therefore, more sensitive volume computation can be achieved by increasing the number of intervals. Note that, the area for non-circle cross-sections is also computed by the Shoelace formula. The volume of a pyramidal frustum is calculated as follows:

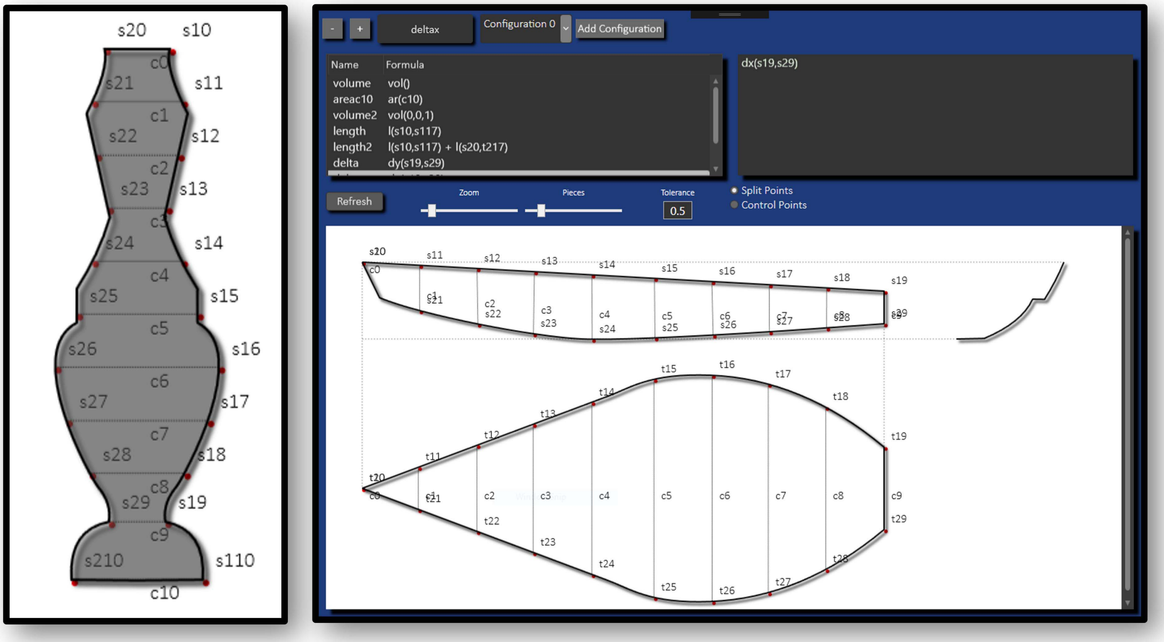


Figure 10: Formula definition interface. Example labeling for the sample with a single view displayed on the left and multi-view on the right. In this interface, the user first defines the number of intervals and points are found on the curve to make sure such intervals. These points on the curve and control points are automatically labelled and displayed to the user. These labels are then used as variables in the functions. In addition, the cross-sections are fit between the points on the curve and labelled to be called by its label. For example, a user can compute the area of the fifth cross-section using the formula $ar(C4)$.

$$V_j = \frac{1}{3} h_j (A_j + A_{j+1} + \sqrt{A_j A_{j+1}})$$

Although there is no doubt for the outcomes of most of the formulas created by such as distance between points, due to shape complexity, the accuracy of the volume calculation over linearly approximated 2D model by intervals may need to be proven. Figure 16 displays 10 vase models and Table 2 displays the calculated volumes using 2D models V^{2D} and 3D models V^{3D}. In the table, the last column refers to error percentage calculated by $100 \times (V^{3D} - V^{2D})/V^{3D}$. According to the results, using the same number of intervals in all 2D models, the error values are ranged between 0.22% and 8.34%. We also applied the ANOVA test and find the p value is < 0.05 and adjusted *R-squared* value as 0.9944 that shows the significant relationships between all the V^{2D} and V^{3D} values. As a result, we can claim that V^{2D} is good enough to estimate the volume of the product and can be used directly in the generative design algorithm that initially produces 2D models.

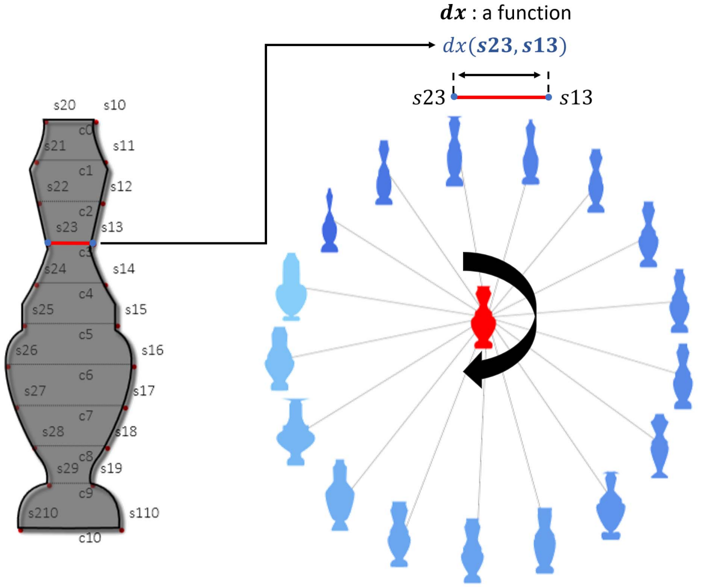


Figure 11: Samples are organized based on an example formula $dx(s23, s13)$ where dx is a function to compute horizontal distance between two points on the profile curves that are labelled as $s23$ and $s13$.

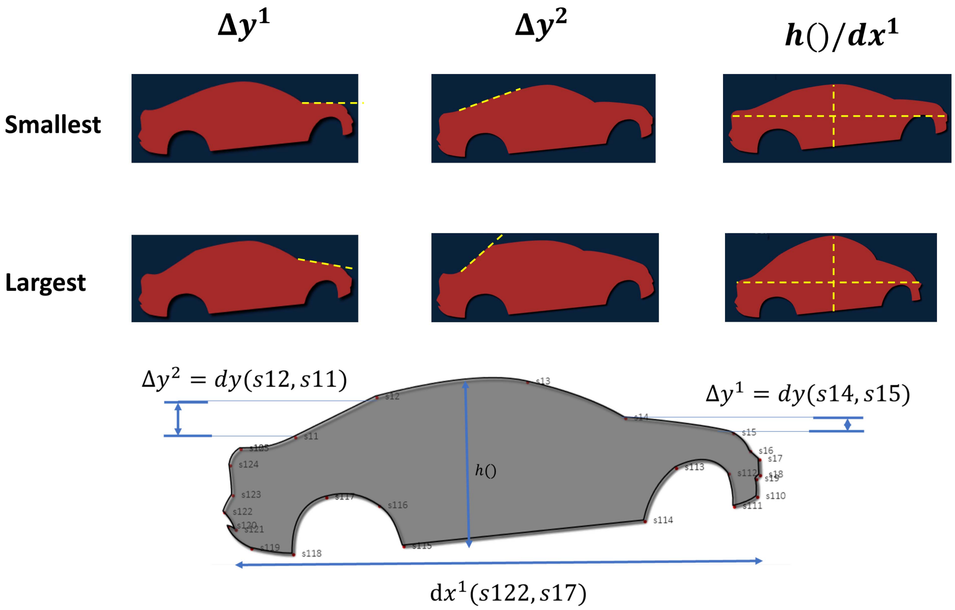


Figure 12: Some geometrical constraint examples for a car side view.

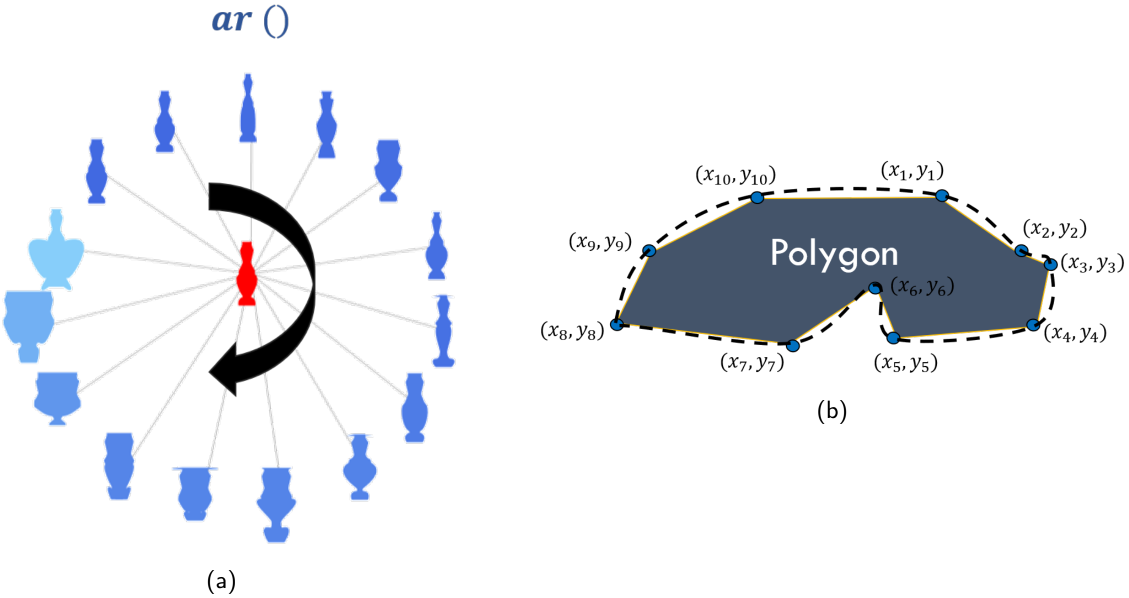


Figure 13: (a) Example implementation of $ar()$ formula that computes side view area of ewer samples. (b) A polygon created by the points on a curve to compute the approximated area.

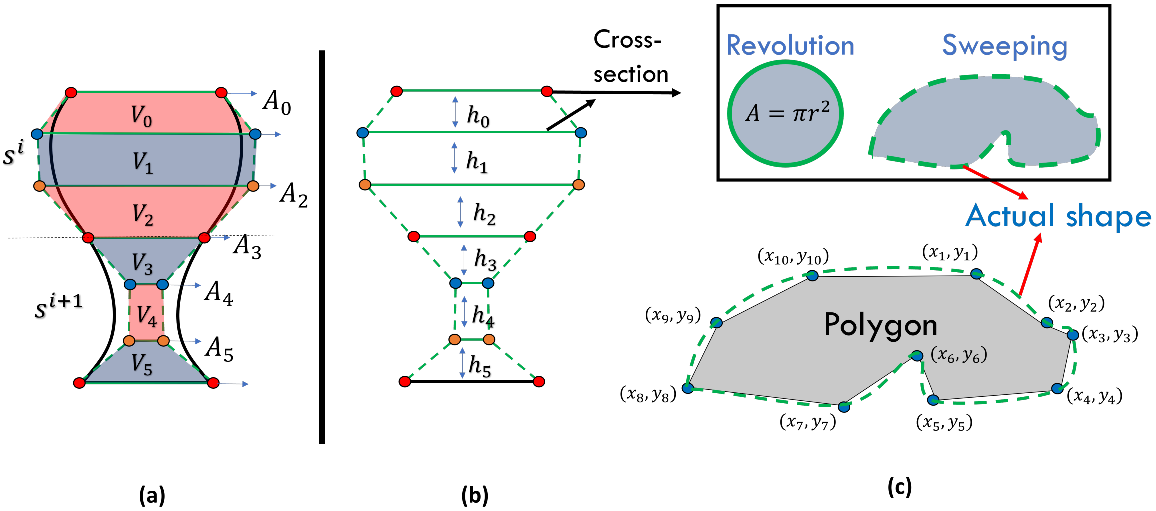


Figure 14: (a) Profile curve with two consecutive segments is represented by black color, and colored regions whose volumes are calculated to show the slices created with control points of the segments. (b) Side of the design indicating the cross-section positions by green lines and the heights between them. (c) Top views of cross-sections when the model is obtained by revolution (circle) or sweeping (polygon).

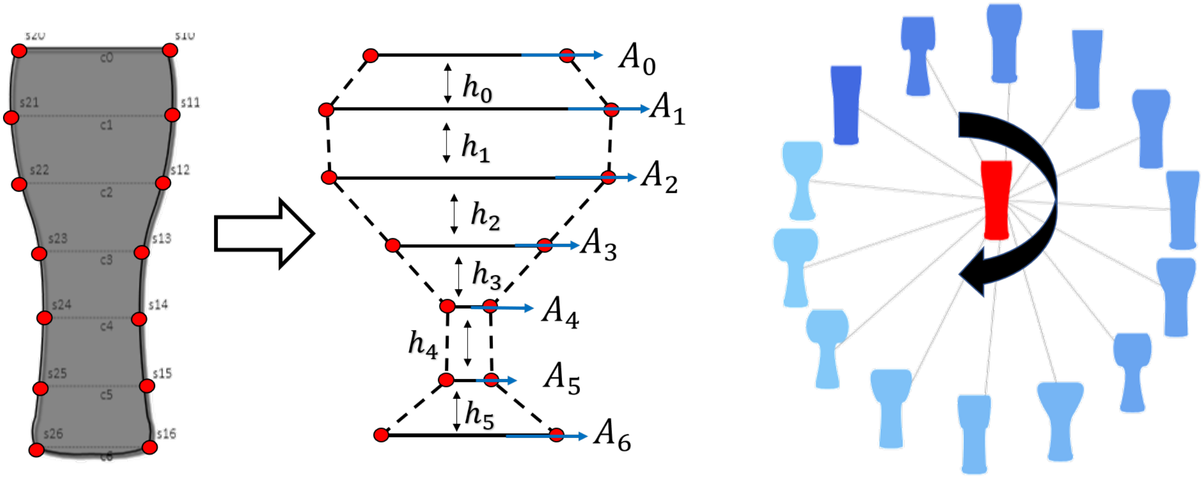


Figure 15: Pyramidal frustums are used to compute volume for each interval and implemented on glass samples using the function vol.

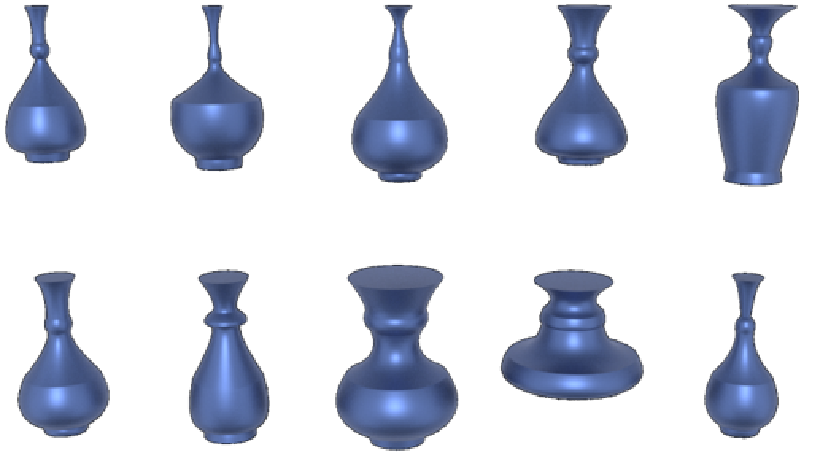


Figure 16: Vase models have been used to compare volumes given in Table 2.

Table 2: Volume calculations over 2D and 3D models.

Design	Vol. calculated from 3D (mm^3)	Vol. calculated from 2D (mm^3)	Error(%)
D1	9 263 500.00	9 091 687.689	1.85
D2	7 642 900.00	7 626 230.861	0.22
D3	9 709 500.00	9 620 120.609	0.92
D4	8 219 000.00	8 100 363.617	1.44
D5	5 364 500.00	5 337 404.956	0.50
D6	5 638 900.00	5 598 396.248	0.72
D7	16 412 000.00	15 042 773.48	8.34
D8	22 983 000.00	22 870 520.82	0.49
D9	9 157 500.00	9 044 061.747	1.24
D10	8 701 700.00	8 647 261.358	0.63

Note that, some functions such as vol, w and h can be computed only for part of the shape defined by segment numbers (see Fig. 17).

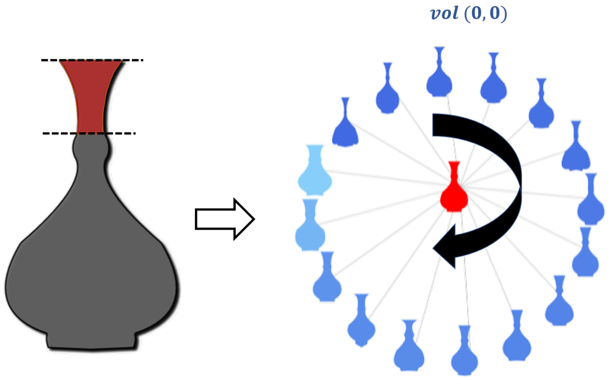
Similarity-based implementation

The similarity constraints can be used for preserving important shape features. As an example, [9] studied an adjective-based design method, where important parameters that make a yacht "charismatic" in the point of view of the customers are determined. Some of the important parameters are displayed in red in the Fig.18, which are D_2^e that is depth of upper curve of the entrance station profile; R_0^m and R_1^m are the minimum radii of curvatures of the relevant guide curves; L_m is the length of the middle section; B_e is the beam of the entrance section. For more information about the parameters and constraints used for hull design generation, we refer to the study of [8]. Note that our aim with the similarity-based implementation is to show how outcomes of various studies can be used along with the system we suggest. Hence, the parameters and how a design is generated are not actually important for the similarity-based implementations. Using such knowledge, new samples are derived from a *charismatic* hull sample for side and top views, which are then combined to generate the displayed 3D models (see Fig. 18). To do that, the design example is first imported and the important parameters for the *charismatic* adjective are kept constant. Therefore, the generated samples are assumed to preserve the prime properties of the *charismatic* hull design. The generated samples are then organized based on anisotropy. The closest and darkest sample is the most similar to the center sample, which is more likely being *charismatic* design. Note that the known important parameters for *charismatic* adjective could also be defined as geometric constraints to achieve the same goals more directly.

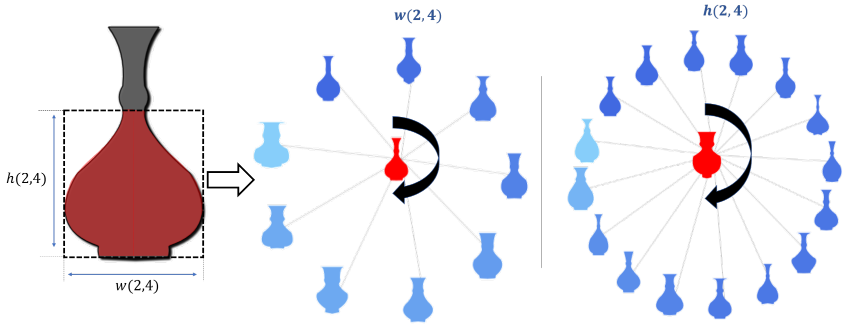
4.2.1 Parametrization

So far the designs are parameterized using the control points of the parametric curves. Even though such approach provides fine level control for the profile modifications and widely used strategy in the literature [14], it becomes unmanageable as the number of parameters increases along with the design complexity.

The proposed method can also be used for an efficient parametrization of the design. To do that, *ud* function called *User Defined* can be utilized to create higher level parametrization where parameters could be length, height and width of the design sections. Figure 19 shows possible parameters on glass example,



(a)



(b)

Figure 17: (a) Volume computation is made only for first segment by $vol(0,0)$. (b) Width and height computation only for the area between 2^{nd} and 4^{th} segments by $w(2,4)$ and $h(2,4)$.

which are $H1, H2, H3, W1, W2, W3$ and $W4$. Table 3 also indicates the relevant formulas to define those parameters.

Table 3: Relevant formulation for the parameters of given design in Figure 19.

Parameter	Function	Sub-formula
H1	ud(dy1)	dy1=dy(s10,s12)
H2	ud(dy2)	dy2=dy(s12,s14)
H3	ud(dy3)	dy3=dy(s14,s16)
W1	ud(dx1)	dx1=dx(s10,s20)
W2	ud(dx2)	dx2=dx(s12,s22)
W3	ud(dx3)	dx3=dx(s14,s24)
W3	ud(dx4)	dx4=dx(s16,s26)

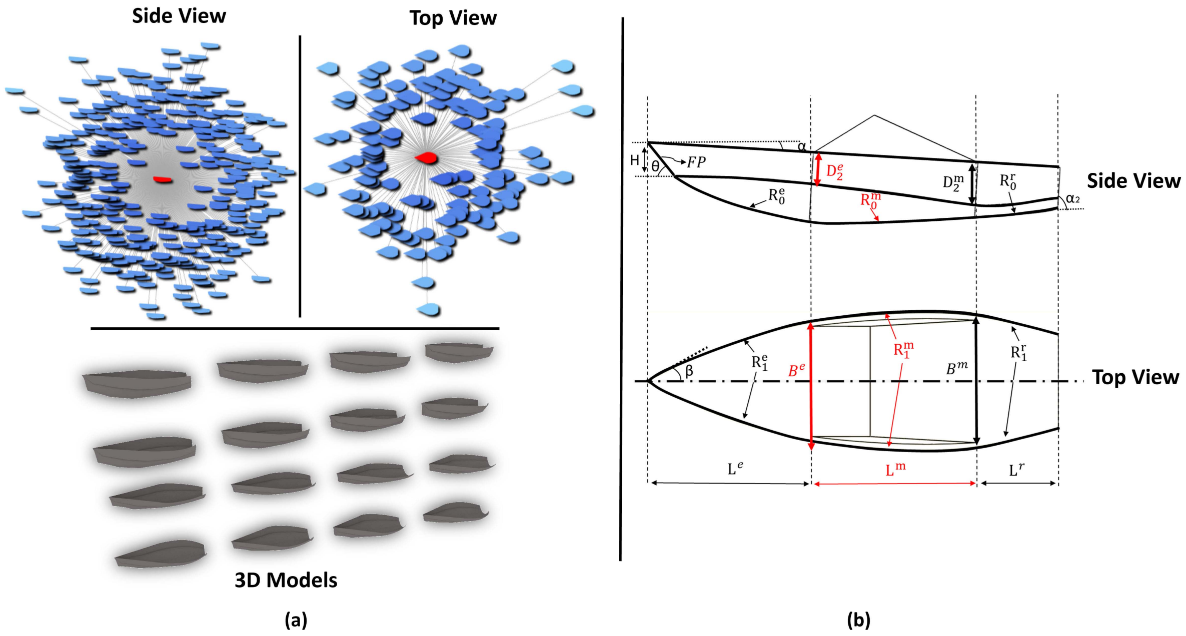


Figure 18: (a) Generated *charismatic* hull samples with side and top views (top), and 3D models (bottom). (b) Illustration of important parameters for *charismatic* adjective of yacht hulls borrowed from the study of [9].

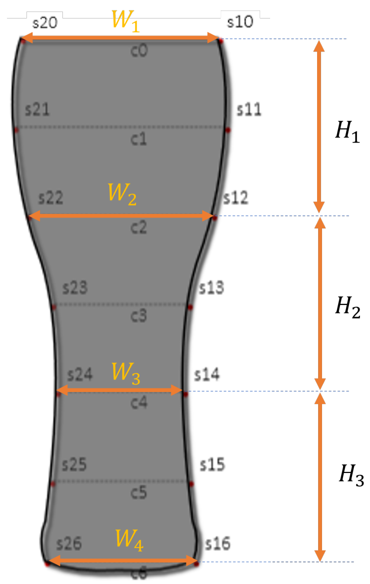


Figure 19: Parameters of simplified parametrization method.

5 Discussion

There is usually a set of geometrical requirements that the product must satisfy aside from the aesthetics defined right at the beginning of the design process. The generated samples can be narrowed down by the proposed user-defined geometric constraint approach, satisfying some of these requirements. For example, a wide bottleneck can be desirable for bottle design in large volumes; therefore, the width of relevant segments can be constrained through the sample management interface. Volume, area and surface area functions can also be used for coarse estimation of the capacity or the required material amount for a product from its design sample. As another example, the ratio length to width and length to height can be used in design when proportions are important. On the other hand, the proposed system is not suitable for constraints that cannot be defined through the profile curves such as surface fairness, the durability of the shape or constraints about manufacturability (i.e., the suitability of a design for injection moulding). Although functional constraints have been discussed mainly, one can also use a mathematical representation of aesthetic definition through the geometrical parameters, which could be obtained human-oriented studies like Kansei Engineering, by converting it into proposed user-defined constraints.

The proposed generative method can be used for various shapes as long as they can be represented by composite Bézier curves with up to 3 views. Extrusion, revolution, sweeping and network surface methods are available to produce 3D models of the profile samples such as hairdryer, yacht hull, ewer, bottle, and glass. For more complicated shapes, like car design, the proposed system can be used for a profile from a single view at a time. On the other hand, since each segment of the profile curve is defined as a Bézier curve, high-quality aesthetic shapes such as those with curvature continuous curves cannot be created by the proposed design scheme. Furthermore, the proposed example based generative design concept can be applied for different shape representation schemes if the shape is represented by parameters, constraining methods are provided to prevent generating irregular shapes, similarity measures are defined, and functions are available to extract some essential conditions from the example design to be conveyed in the generated samples.

In addition to use of control points or points on curves as parameters, our system can adopt various parameterization techniques proposed in the literature by converting the relevant parameters into sub-formulas through *ud* function. As an example, [19] proposed a novel design parameterization technique for yacht hulls, in which the design is divided into three regions with parameters such as beam, length, depth and various angles that are important for hydro-static performance of the hulls. As the corresponding parameters of their parameters can be defined with our approach, we can examine the shape properties by means of mathematical formulas that [19] have not tried on the generated designs.

6 Conclusion and Future Works

Industrial and engineering designers often work together in the conceptual design stage to find solutions for determining design problems. They sketch their solution ideas in 2D basic forms for a design's various views. When creating such sketches, the designers often are acknowledged basic essential geometrical constraints that a product must have because of specific functional requirements. To find the optimal design, generative design is a good practice since it creates many design alternatives using computers, which increase the chance of exploring more promising solution candidates in a shorter time. Most of the generative design methods, including the one adopted in this study using one or more initial designs to generate new design ideas, as designers normally do, to imagine how could new designs be innovative. Without a doubt, computers can do better if it is used efficiently as it is free from prejudgement and more capable in terms of computational abilities. Earlier proposed example based generative design method [10] make it easier to use computers to generate the design samples. However, the system is incapable of covering functional constraints, which limits its usage by only finding a variety of designs in terms of aesthetics.

The proposed interface and the study aim to improve a design's 2D views separately before its 3D form is generated. It is believed such an approach is useful for the conceptual design stage, specifically when only a photo or sketch of a design is available rather than a 3D CAD model. In future studies, the system will also support statistical methods to predict outcomes using simple built-in functions. For example, [13] use car side silhouettes to predict the aerodynamic performance of the car designs employing a machine learning technique. We can let users upload a prediction model obtained using a machine learning technique to our system to predict relevant outcomes of the designs to be managed in the sample management interface. On the other hand, we also aim to extend our approach to make it applicable directly on a 3D model so that we can assist the designers in later stages, such as embodiment design. Furthermore, virtual reality integration into the proposed system is thought of as one of the future works of this study. Thanks to this, it is believed the virtual prototyping used for visual communication between the design creator and others can replace the need to create physical prototypes for immature design ideas.

ACKNOWLEDGEMENTS

The major part of the work of this paper was conducted as the first author's Ph.D. thesis project at The University of Tokyo. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

ORCID

Kemal M. Dogan, http://orcid.org/0000-0003-1015-7266
Hiromasa Suzuki, http://orcid.org/0000-0003-4237-1101

REFERENCES

[1] Alcaide-Marzal, J.; Diego-Más, J.A.; Asensio-Cuesta, S.; Piqueras-Fiszman, B.: An exploratory study on the use of digital sculpting in conceptual product design. Design Studies, 34(2), 264–284, 2013. http://doi.org/10.1016/j.destud.2012.09.001.

[2] Bao, Q.; Faas, D.; Yang, M.: Interplay of sketching & prototyping in early stage product design. International Journal of Design Creativity and Innovation, 6(3-4), 146–168, 2018. http://doi.org/10.1080/21650349.2018.1429318.

[3] Bilda, Z.; Gero, J.S.; Purcell, T.: To sketch or not to sketch? that is the question. Design studies, 27(5), 587–613, 2006. http://doi.org/10.1016/j.destud.2006.02.002.

[4] Bookstein, F.L.: Size and shape spaces for landmark data in two dimensions. Statistical science, 181–222, 1986. http://doi.org/10.1214/ss/1177013696.

[5] Chu, C.H.; Song, M.C.; Luo, V.C.: Computer aided parametric design for 3d tire mold production. Computers in Industry, 57(1), 11–25, 2006. http://doi.org/10.1016/j.compind.2005.04.005.

[6] Cluzel, F.; Yannou, B.; Dihlmann, M.: Using evolutionary design to interactively sketch car silhouettes and stimulate designer's creativity. Engineering Applications of Artificial Intelligence, 25(7), 1413–1424, 2012. http://doi.org/10.1016/j.engappai.2012.02.011.

[7] Deng, Y.M.; Tor, S.B.; Britton, G.: Abstracting and exploring functional design information for conceptual mechanical product design. Engineering with computers, 16(1), 36–52, 2000. http://doi.org/10.1007/s003660050035.

[8] Dogan, K.M.; Gunpinar, E.: Learning yacht hull adjectives and their relationship with hull surface geometry using gmdh-type neural networks for human oriented smart design. Ocean Engineering, 145, 215–229, 2017. http://doi.org/10.1016/j.oceaneng.2017.08.056.

[9] Dogan, K.M.; Suzuki, H.; Gunpinar, E.: Eye tracking for screening design parameters in adjective-based design of yacht hull. Ocean Engineering, 166, 262–277, 2018. http://doi.org/10.1016/j.oceaneng.2018.08.026.

[10] Dogan, K.M.; Suzuki, H.; Gunpinar, E.; Kim, M.S.: A generative sampling system for profile designs with shape constraints and user evaluation. Computer-Aided Design, 111, 93–112, 2019. http://doi.org/10.1016/j.cad.2019.02.002.

[11] Dorst, K.; Cross, N.: Creativity in the design process: co-evolution of problem–solution. Design studies, 22(5), 425–437, 2001. http://doi.org/10.1016/S0142-694X(01)00009-6.

[12] Dubuisson, M.P.; Jain, A.K.: A modified hausdorff distance for object matching. In Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on, vol. 1, 566–568. IEEE, 1994. http://doi.org/10.1109/ICPR.1994.576361.

[13] Gunpinar, E.; Coskun, U.C.; Ozsipahi, M.; Gunpinar, S.: A generative design and drag coefficient prediction system for sedan car side silhouettes based on computational fluid dynamics. Computer-Aided Design, 111, 65–79, 2019. ISSN 0010-4485. http://doi.org/10.1016/j.cad.2019.02.003.

[14] Gunpinar, E.; Gunpinar, S.: A shape sampling technique via particle tracing for CAD models. Graphical Models, 96, 11–29, 2018. ISSN 1524-0703. http://doi.org/10.1016/j.gmod.2018.01.003.

[15] Hartigan, J.A.; Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100–108, 1979. http://doi.org/10.2307/2346830.

[16] Kazi, R.H.; Grossman, T.; Cheong, H.; Hashemi, A.; Fitzmaurice, G.W.: Dreamsketch: Early stage 3d design explorations with sketching and generative design. In UIST, vol. 14, 401–414, 2017. http://doi.org/10.1145/3173574.3173943.

[17] Kelly, J.; Papalambros, P.Y.; Seifert, C.M.: Interactive genetic algorithms for use as creativity enhancement tools. In AAAI spring symposium: creative intelligent systems, 34–39, 2008.

[18] Khan, S.; Gunpinar, E.: Sampling CAD models via an extended teaching–learning-based optimization technique. Computer-Aided Design, 100, 52–67, 2018. http://doi.org/10.1016/j.cad.2018.03.003.

[19] Khan, S.; Gunpinar, E.; Dogan, K.M.: A novel design framework for generation and parametric modification of yacht hull surfaces. Ocean Engineering, 136, 243–259, 2017. http://doi.org/10.1016/j.oceaneng.2017.03.013.

[20] Kim, K.; Lee, K.p.: Collaborative product design processes of industrial design and engineering design in consumer product companies. Design Studies, 46, 226–260, 2016. http://doi.org/10.1016/j.destud.2016.06.003.

[21] Koyama, Y.; Sakamoto, D.; Igarashi, T.: Crowd-powered parameter analysis for visual design exploration. In Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology, UIST '14, 65–74. Association for Computing Machinery, New York, NY, USA, 2014. ISBN 9781450330695. http://doi.org/10.1145/2642918.2647386.

[22] Krish, S.: A practical generative design method. Computer-Aided Design, 43(1), 88–100, 2011. http://doi.org/10.1016/j.cad.2010.09.009.

[23] Krtolica, P.V.; Stanimirović, P.S.: Reverse polish notation method. International Journal of Computer Mathematics, 81(3), 273–284, 2004. http://doi.org/10.1080/00207160410001660826.

[24] Kudrowitz, B.; Te, P.; Wallace, D.: The influence of sketch quality on perception of product-idea creativity. AI EDAM, 26(3), 267–279, 2012. http://doi.org/10.1017/S0890060412000145.

[25] Lee, Y.; Lim, W.: Shoelace formula: Connecting the area of a polygon and the vector cross product. The Mathematics Teacher, 110(8), 631–636, 2017. http://doi.org/10.5951/mathteacher.110.8.0631.

[26] Matejka, J.; Glueck, M.; Bradner, E.; Hashemi, A.; Grossman, T.; Fitzmaurice, G.: Dream lens: Exploration and visualization of large-scale generative design datasets. In Proceedings of the 2018 CHI conference on human factors in computing systems, 1–12, 2018.

[27] McCormack, J.; Dorin, A.; Innocent, T.; et al.: Generative design: a paradigm for design research. Proceedings of Futureground, Design Research Society, Melbourne, 2004. https://dl.designresearchsociety.org/drs-conference-papers/drs2004/researchpapers/171.

[28] McNeel, R.: Grasshopper-generative modeling with rhino, mcneel north america, seattle, usa, 2010.

[29] Mountstephens, J.; Teo, J.: Progress and challenges in generative product design: A review of systems. Computers, 9(4), 80, 2020. http://doi.org/10.3390/computers9040080.

[30] Orbay, G.; Ersin Yumer, M.; Kara, L.B.: Sketch-based aesthetic product form exploration from existing images using piecewise clothoid curves. Journal of Visual Languages & Computing, 23(6), 327–339, 2012. ISSN 1045-926X. http://doi.org/10.1016/j.jvlc.2012.07.001.

[31] orcasquall: Reverse polish notation in C#, 2007. https://www.dreamincode.net/forums/topic/35320-reverse-polish-notation-in-c%23/. [Online; posted 14-October-2007].

[32] Orsborn, S.; Cagan, J.; Pawlicki, R.; Smith, R.C.: Creating cross-over vehicles: Defining and combining vehicle classes using shape grammars. Ai Edam, 20(3), 217–246, 2006. http://doi.org/10.1017/S0890060406060185.

[33] Pahl, G.; Beitz, W.: Engineering design: a systematic approach. Springer Science & Business Media, 2013. ISBN 978-1846283185.

[34] Pugliese, M.J.; Cagan, J.: Capturing a rebel: modeling the harley-davidson brand through a motorcycle shape grammar. Research in Engineering Design, 13(3), 139–156, 2002. http://doi.org/10.1007/s00163-002-0013-1.

[35] Rahimian, F.P.; Ibrahim, R.: Impacts of vr 3d sketching on novice designers' spatial cognition in collaborative conceptual architectural design. Design Studies, 32(3), 255–291, 2011. http://doi.org/10.1016/j.destud.2010.10.003.

[36] Shih, Y.T.; Sher, W.D.; Taylor, M.: Using suitable design media appropriately: Understanding how designers interact with sketching and CAD modelling in design processes. Design studies, 53, 47–77, 2017. http://doi.org/10.1016/j.destud.2017.06.005.

[37] Singh, V.; Gu, N.: Towards an integrated generative design framework. Design studies, 33(2), 185–207, 2012. http://doi.org/10.1016/j.destud.2011.06.001.

[38] Sponberg, E.W.: The design ratios. Sponberg Yacht Design Inc, 2011.

[39] Sturges, R.H.; O'Shaughnessy, K.; Kilani, M.I.: Computational model for conceptual design based on extended function logic. AI EDAM, 10(4), 255–274, 1996. http://doi.org/10.1017/S089006040000161X.

[40] Ulrich, K.T.; Eppinger, S.D.: Product design and development. McGraw-Hill New York, 1995. ISBN 978-0078029066.

[41] Valencia-Romero, A.; Lugo, J.E.: Part-worth utilities of gestalt principles for product esthetics: a case study of a bottle silhouette. Journal of Mechanical Design, 138(8), 081102, 2016. http://doi.org/10.1115/1.4033664.

[42] Vuletic, T.; Duffy, A.; Hay, L.; McTeague, C.; Pidgeon, L.; Grealy, M.: The challenges in computer supported conceptual engineering design. Computers in Industry, 95, 22–37, 2018. http://doi.org/10.1016/j.compind.2017.11.003.