Proceedings of the ASME 2009 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2009
August 30-September 2, 2009, San Diego, USA

DETC2009-87477

LEARNING GEOMETRIC DESIGN KNOWLEDGE FROM CONCEPTUAL SKETCHES AND ITS UTILIZATION IN SHAPE CREATION AND OPTIMIZATION

Gunay Orbay
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: gorbay@andrew.cmu.edu

Levent Burak Kara
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
Email: lkara@andrew.cmu.edu

ABSTRACT

In current product design, significant effort is put into creating aesthetically pleasing product forms. Often times, the final shape evolves in time based on designers' ideas externalized through early design activities primarily involving conceptual sketches. While designers negotiate and convey a multitude of different ideas through such informal activities, current computational tools are not well suited to work from such forms of information to leverage downstream design processes. As a result, many promising ideas either remain under-explored, or require restrictive added effort to be transformed into digital media. As one step toward alleviating this difficulty, we propose a new computational method for capturing and reusing knowledge regarding the shape of a developing design from designers' hand-drawn conceptual sketches. At the heart of our approach is a geometric learning method that involves constructing a continuous space of meaningful shapes via a deformation analysis of the constituent exemplars. The computed design space serves as a medium for encoding designers' shape preferences expressed through their sketches. With the proposed approach, designers can record desirable shape ideas in the form of raw sketches, while utilizing the accumulated information to create and explore novel shapes in the future. A key advantage of the proposed system is that it enables prescribed engineering and ergonomic criteria to be concurrently considered with form design, thus allowing such information to suitably guide conceptual design processes in a timely manner.

INTRODUCTION

Creating aesthetic product forms is a key challenge in current industrial design and product development environments. In early design phases, designers put significant effort into exploring a multitude of different ideas regarding product form through rough conceptual sketches [1]. Such sketches embody key knowledge regarding designers' evolving concepts, and cumulatively serve as a library of seed ideas that guides the development of the final shape. Several studies have shown that the quality and quantity of such early sketches have a pivotal impact on the design outcome [2–5]. Nonetheless, a key drawback in the current design practice is that the information arising through such activities cannot be easily communicated to downstream design processes with conventional computational tools, resulting in many promising concepts to remain under-explored or abandoned prematurely. Additionally, the lack of means to digitally encode and communicate such information causes multiple iterations between design and engineering teams [6]. As a consequence, the final product form usually evolves in time until a compromise between designers' conceptual designs and the underlying engineering and ergonomics requirements is met. Despite this practice, current computational tools provide little or no support to capture geometric design knowledge from designers' conceptual sketches and to reuse it throughout the product design process.

In this work, we propose a new computational method for capturing and reuse geometric design information from designers' conceptual sketches. We achieve this by defining a continu-

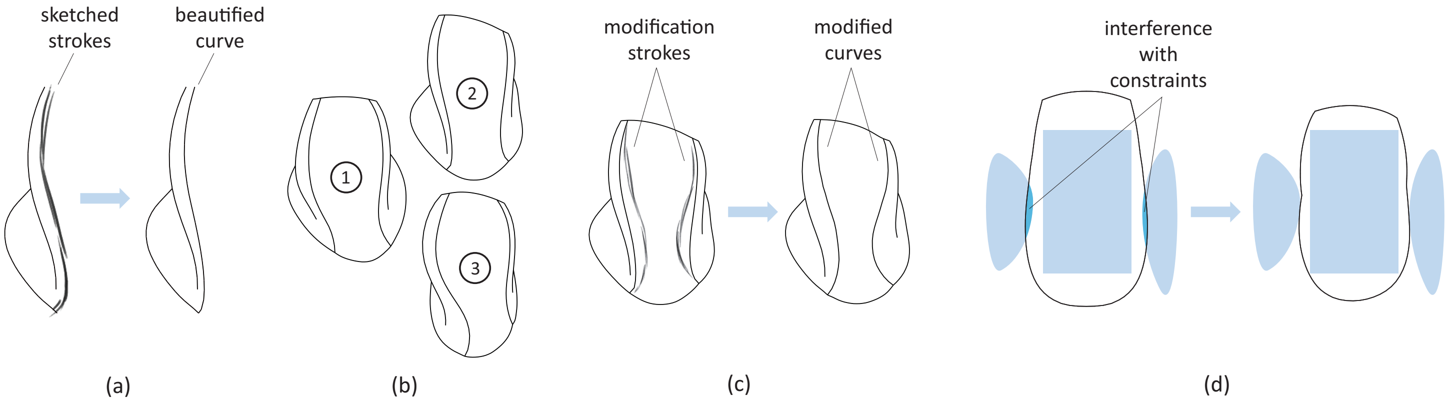

Figure 1. Demonstration of the proposed method including (a) sketching ideas via a sketch-based design interface, (b) constructing a design space with multiple sketches, (c) sketching over an existing design for exploration, and (d) shape optimization under engineering and ergonomics constraints.

ous space of meaningful shapes which are derived via geometric deformations between constituent exemplars. By this example-based approach, the style and aesthetics knowledge inherent in design sketches is implicitly identified and reused. This design space allows designers to interactive explore, populate and evolve new designs all though a sketch-based design interface. It also enables integrated shape optimization to look for solutions to underlying engineering and ergonomics constraints. Doing so, our method brings engineering and ergonomics considerations in to the early conceptual design. Our method is currently 2D.

RELATED WORK

Developing technology in support of conceptual design activities has been an active area of research for both industrial and engineering design [7–16]. Most approaches aim to generate new designs based on a corpus of existing design solutions, but typically require explicit parameterization or prescribed generative rules. Genetic Algorithms (GA), for instance, have been a popular choice of generative design in many domains including product design [7–10] and engineering design [11]. Generated designs are evaluated either by the user (*i.e.* aesthetic judgment of the designer) or by predefined evaluation functions. In addition, shape grammars have been used to generate new designs by defining a template topology based on geometric rules [12–14]. Example-based exploration methods have also found applications in automotive industry [15,16]. In all the above approaches, a suitable parameterization of the problem has to be established before exploring design alternatives. Often times, the quality of the parameter choice dictates the variety of the design space from which new designs can be derived. Moreover, this parameterization typically cannot be automatically initiated from the set of provided design solutions.

Recently, there has been an increasing interest in learning designers' preferences through their interactions with computational tools. Moss *et al.* [17] studied learning "chunks of information" from previous designs and used this information to generate new designs in an agent-based design system. They demonstrated the learning capability by achieving more acceptable designs with less number of iterations. Likewise, Kurtoglu and Campbell [18] developed a "design preference modeler" for the evaluation of automatically generated designs. Wannarumon *et al.* [10] compiled a general aesthetics model from aesthetics measures defined on geometric properties (*e.g.* symmetry, golden ratio), and consumer responses. They suggested that with a similar approach designers' aesthetic preferences can be learned statistically. Although the above approaches are capable of learning aesthetics and engineering preferences from experience, they are not suitable for extracting and reusing geometric design knowledge directly from sketches.

In the last two decades, there has been a significant interest toward utilizing sketch input via pen-based computer interfaces. With new 3D geometry construction and deformation methods, creation of different types of curves and surfaces are now possible via free hand sketching. Earlier studies were focused on creating primitive 3D geometries by using a few simple strokes [19, 20] while recent studies present novel free-form surface creation methods [21–24]. Although useful in their target domains, these methods do not readily extend to *learning* geometric design knowledge.

Above approaches provide assistance in product design process by producing design alternatives, by learning design preferences from experience, and by forming 3D product geometries from sketch input. However, these tools are not capable of learning design information inherent in conceptual design sketches and reusing it throughout the design process. In this work, we propose a method to support styling design of industrial products by learning geometric design knowledge from a set of conceptual design sketches and reusing it in shape creation and optimization under engineering and ergonomics constraints.

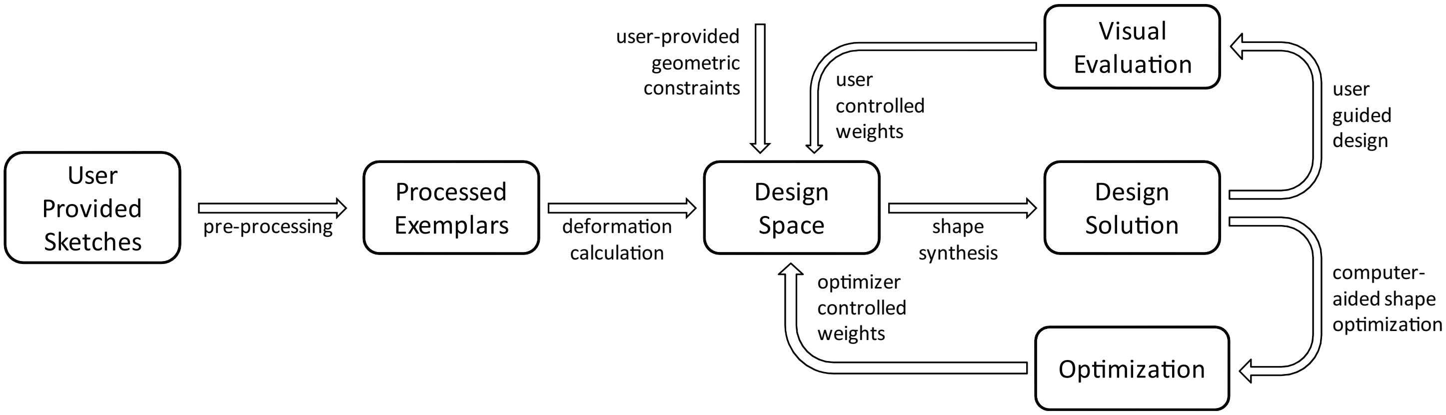

Figure 2. Proposed method initiates with user-provided conceptual sketches. It then develops these sketches into a design space. Design space is either manually explored or used in integrated shape optimization.

TECHNICAL APPROACH

Overview of the Method

The proposed exemplar-based method provides computational assistance for early design stages by enabling designers to record and reuse their emerging ideas involving form. Our approach is based on constructing and managing a geometric *design space* that originates from designers' input sketches. Once created, the design space allows designers to explore novel product forms commensurate with the input exemplars, while allowing prescribed engineering and ergonomic constraints to be concurrently satisfied via integrated optimization.

Figure 2 outlines the main steps of our method. In a typical scenario, users sketch a number of alternative concepts for the problem at hand using a pen-based computer interface. The constituent strokes of the input sketches are first transformed into beautified curves, forming a suitable mathematical representation of the input sketches for subsequent geometric analysis. Each beautified sketch defines a discrete *exemplar* in the evolving design space that spans a continuous space of admissible designs. From the constructed design space, new solutions can be synthesized either manually through user-guided shape exploration, or automatically through computer controlled optimization.

Our design space based approach enables designers to:

1. Modify a working design through sketching while the method automatically explores the design space according to designers' modifications.
2. Modify a working design through sketching while they manually explore the design space.
3. Define engineering and ergonomics objectives and let the integrated optimization process determine a solution from the design space.
4. Populate the design space with the synthesized designs.

At its core, our method is based on the feature space formulation introduced by Sumner *et al.* [25]. While this formulation calculates continuous transitions between different geometries, it also allows shapes to flexibly take form under geometric constraints while undergoing minimal local deformations. Thus, the resulting geometries are not limited to provided exemplars, instead it can flexibly produce new geometries in accordance with constraints and constituent geometries. Moreover, in cases where large amount of rotational deformation is required, it can still produce meaningful transitions by treating rotations separately from other types of deformations.

Geometric Representation

Most conceptual design sketches consist of pen strokes, colors and marker shadings that convey the geometry and texture of an underlying object, which can be easily interpreted by humans. However, performing the same interpretation by computers is not as easy. To facilitate the proposed work, we assume sketches to involve only contours, edges and important character curves without shading and coloring. With this, a sketch can be modeled as a combination of simple geometries, including multiple line segments and/or parametric curves. In the following section, we only give formulation for line segment representation, but it can be extended to parametric curves without much effort.

Often times, designers use a series of multiple strokes to express a particular curve. In such cases, the strokes recorded by a graphics tablet, are converted (*i.e.* beautified) into a chain of line segments through a series of smoothing, ordering and resampling operations as proposed in [24]. A sample case is given Figure 3.(a,b) to demonstrate the conversion from multiple pen strokes to beautified curves.

The proposed method is based on calculating deformation gradients between line segments of different geometries. Since deformation is a one to one matching, it is crucial to have consis-

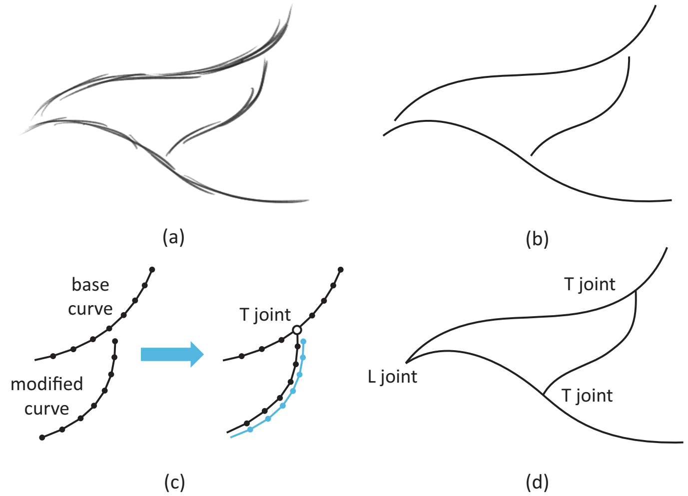

Figure 3. Geometric representation of sketches: (a) Hand drawn sketch strokes (b) are beautified into multiple line segments. (c) T and L types of joints are defined by the user (d) to finalize geometric representation.

tent topologies for all geometries. Additionally, it is important to know which part of a sketch corresponds to which part of another in terms of geometric and functional properties. An example is shown in Figure 4. Although correspondence between end points is evident, it is not clear where the sharp corner on the left hand side curve corresponds to on the right hand side curve. Since this correspondence will depict how the transition between geometries would occur, it is crucial to define it. In the following sections, we assume that the user provides the correspondence information by sketching consistent shapes.

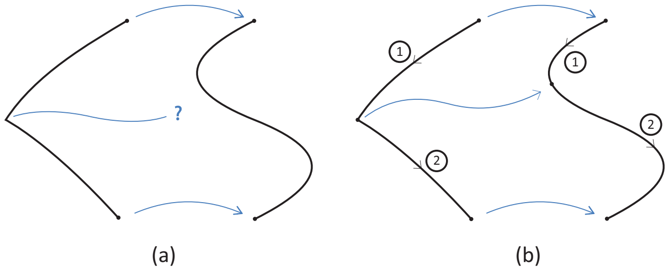

Figure 4. Correspondence issue: Example in (a) illustrates possible ambiguities in correspondence between sketches. In such situations, (b) designer provides additional information by drawing curve segments in consistent drawing ordering and direction. (Dots denote end points)

Design Space Creation

Defining a continuous design space from a number of sketches requires calculating intermediate shapes which lie between provided exemplars. By using the above graphical representation, this task is reduced to transforming one geometry into another. The “Mesh-Based Inverse Kinematics" (MeshIK) formulation by Sumner *et al.* [25] satisfies this need while introducing a flexible deformation that allows multiple geometric constraints. The original purpose of their work is calculating meaningful poses for given constraints by using a set of given seed poses for inverse kinematics purposes. Therefore, their formulation is sensitive to overall scale of constraints as well as their relative positions. On the contrary, we consider that shapes and proportions define the qualitative properties of a sketch irrespective of its overall size. Also, it is not mandatory and guaranteed that all the provided examples are of the same scale. Therefore, we modified the formulation to make it invariant to size. In this section, a brief formulation for MeshIK will be given for 2D space. For more information please see reference [25]. A similar formulation was also given by Kókai *et al.* in [16].

Formulation A seemingly straightforward approach to constructing a design space from a set of given exemplars is to represent a new shape as a linear combination of global coordinate vectors of each exemplar. Although theoretically viable, a linear combination is generally poor in blending local features having large deformations among exemplars. Instead, new shapes can be defined by a combination of vectors storing deformations gradient matrices, namely feature vectors. New geometries can then be generated by using the resulting combination of the feature vectors.

In our formulation the design space is spanned by the deformations between a reference sketch and other sketches. Assuming each sketch is composed of n number of vertices and m number of line segments, we would like to calculate the deformation gradient for each line segment in the reference sketch S_0 and the corresponding line segment in another sketch, S. But, for line segments in 2D space, there is no unique solution for the deformation gradient (*i.e.* two points can be mapped onto other two points in two ways). To come up with a unique transformation it is required to define another point for each line segment to form a triangle. The third vertex of j^{th} line segment can be defined by rotating the line around its first vertex by 90 degrees, as shown in Figure 5.a. The coordinates of the third vertex are calculated as follows:

$$\mathbf{v}_3^j = \mathbf{v}_1^j + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} (\mathbf{v}_2^j - \mathbf{v}_1^j) \qquad (1)$$

where $\mathbf{v}$ denotes the position vector that stores the x and y coordinates of a vertex. After defining a third vertex for m such segments, the total number of vertices increases to $n+m$. Once all the segments are converted into triangles, an affine transformation that maps the j^{th} triangle in the reference geometry on to the corresponding triangle in the deformed geometry is defined (Fig. 5.b).

$$\Phi_j(\mathbf{p}) = \mathbf{T}_j\mathbf{p} + \mathbf{t}_j \tag{2}$$

Here, $\mathbf{T}_j$ is a 2×2 matrix including rotation and scale type deformations and $\mathbf{t}_j$ is a vector that provides translation component of the affine transformation. To determine $\mathbf{T}_j$ in Equation 2, the translation component must be eliminated. This is achieved by subtracting the equation written for the third vertex, from the equations written for the original two vertices. From the two remaining equations, the transformation matrix is determined as follows:

$$\mathbf{T}_j = [\,\mathbf{v}_1^j - \mathbf{v}_3^j \;\; \mathbf{v}_2^j - \mathbf{v}_3^j\,].[\,\bar{\mathbf{v}}_1^j - \bar{\mathbf{v}}_3^j \;\; \bar{\mathbf{v}}_2^j - \bar{\mathbf{v}}_3^j\,]^{-1} \tag{3}$$

where $\mathbf{v}$ and $\bar{\mathbf{v}}$ denote coordinate vectors of the deformed and reference geometries, respectively. It should be noted that $\mathbf{T}_j$ is linear in the coordinates of the deformed sketch S. Thus, a linear relationship between coordinates of deformed geometry S and a governing feature vector $\mathbf{f}$ can be written as

$$\mathbf{f} = \mathbf{G}\mathbf{x} \tag{4}$$

where $\mathbf{x}$ is a vector storing the global coordinates of the deformed sketch in the form of $x = [x_1, x_2, x_3 \ldots y_1, y_2, y_3 \ldots]^T$, $\mathbf{f}$ is the feature vector and $\mathbf{G}$ is a sparse block diagonal matrix whose coefficients only depend on the coordinates of the original sketch S_0. Feature vector $\mathbf{f}$ is the deformation that defines geometry S with respect to a reference geometry S_0 whereas $\mathbf{G}$ can be interpreted as the mapping between an arbitrary geometry and its associated feature vector defined with respect to S_0.

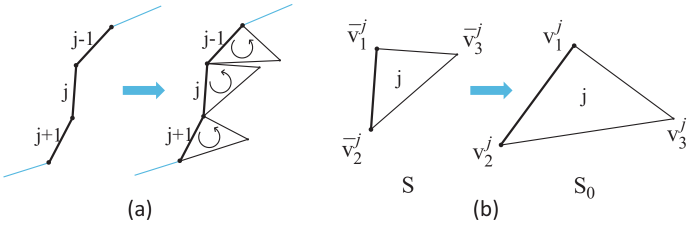

Figure 5. Transformation between exemplars: (a) Multiple line segment representation of sketches are first pre-processed by defining a third vertex for each segment. (b) j^{th} triangle in S_0 then undergoes an affine transformation to yield the corresponding triangle in S

Given a feature vector, the global coordinates can be calculated by inverse operating $\mathbf{G}$ on $\mathbf{f}$. However, since $\mathbf{G}$ was constructed without the translation component of the affine transformation, coordinates of at least one point must be specified to anchor the shape. The global coordinates of the remaining vertices are then determined by the following minimization expression

$$\mathbf{x} = \arg\min_{\mathbf{x}} \| \overline{\mathbf{G}}\mathbf{x} - (\mathbf{f} - \mathbf{c}) \| \tag{5}$$

where $\mathbf{c}$ is the vector obtained by multiplying the constraint coordinates with the associated columns of $\mathbf{G}$. $\mathbf{c}$ can also be considered as a feature vector that forces free coordinates to align with given constraints. Moreover, $\overline{\mathbf{G}}$ is matrix $\mathbf{G}$ without the columns associated with the constrained coordinates. In case of multiple constraints, solving Equation 5 yields the most meaningful shape which results in minimum amount of local deformations. In practice, the constraints on the geometry are defined by drawing modification strokes as shown in Figure 1.c.

Design Space Once all feature vectors are calculated for l number of sketches, a design space can be defined by a weighted average of the feature vectors $\mathbf{f}_i$ using weights w_i, which determine the contributions of each exemplar on the resulting shape, as follows

$$\mathbf{f}_\mathbf{W} = \sum_{i=1}^{l} w_i \mathbf{f}_i \tag{6}$$

Once again, linear combinations of feature vectors result in weak blending in cases where large deformations and rotations are involved. Hence, the rotation and scaling part of the transformation matrix are combined separately. Transformation matrix $\mathbf{T}_j$ of the j^{th} line segment can be decomposed into rotation ($\mathbf{R}_j$) and scaling ($\mathbf{S}_j$) parts into the following form via polar decomposition [26]:

$$\mathbf{T}_j = \mathbf{R}_j \mathbf{S}_j \tag{7}$$

Once the rotation part is determined, the angle traversed can be calculated by using matrix logarithm. With this decomposition, the combination of feature vectors can be carried out in angle and scaling level separately. Upon combination, the feature vector is calculated using matrix exponential as follows:

$$\mathbf{T}_j(\mathbf{w}) = \exp\left(\sum_{i=1}^{l} w_i \log(\mathbf{R}_{ij})\right) . \sum_{i=1}^{l} w_i \mathbf{S}_{ij} \tag{8}$$

The utility of non-linear design space representation is illustrated in Figure 6. In addition to generating shapes in-between exemplars, it can readily extrapolate to deformations involving large rotations. For both linear and nonlinear formulation of the design space, the transformation matrices are insensitive to translations. Thus, calculating the global coordinates using a feature vector requires the coordinates of at least one point to be defined. Then the new shape can be calculated by solving the following minimization problem:

$$\mathbf{x} = \arg\min_{\mathbf{x},\mathbf{w}} \|\overline{\mathbf{G}}\mathbf{x} - (\mathbf{M}(\mathbf{w}) - \mathbf{c})\| \tag{9}$$

where $\mathbf{M}(\mathbf{w})$ is the feature vector as a function of weights $\mathbf{w}$. This expression not only minimizes the total amount local deformations but also automatically adjusts weights to yield the most meaningful shape that can be achieved from the provided geometries.

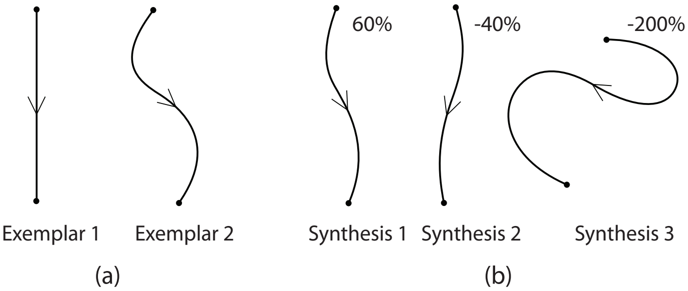

Figure 6. Shape synthesis using non-linear design space: (a) While deformation from first exemplar to second (b) can be interpolated with either (left) positive or (middle) negative amounts, non-linear design space can also handle (right) extrapolations involving high deformations

Scale Invariance As mentioned before, sketches contain geometric information related to an underlying design and the overall impression of that design does not depend on its scale, but rather its *shape*. Likewise, designers sketch their concepts without providing information on exact dimensions of their designs. Therefore, we introduced modifications on the original feature space formulation to make it invariant to scale. For this, we first normalize the scaling component of deformation gradient for each sketch with respect to the reference sketch as follows:

$$\psi_i' = \psi_i \frac{\|\mathbf{I}\|}{\|\psi_i\|} \tag{10}$$

where ψ_i' and ψ_i are the normalized and original scaling feature vectors, which are constructed the same way features vectors are constructed, $\mathbf{I}$ is the scale feature vector which is constructed from identity transformation matrices , and $\|.\|$ is the matrix norm operator.

Although we normalize each design to a compatible scale, the minimization expression in Equation 9 does not necessarily provide scale invariance since it lacks a free scaling parameter. Including the reference sketch as a deformed sketch would provide unity transformation matrices for each line segment, thus its associated weight becomes the free scaling parameter that can be adjusted during the minimization process of Equation 9. Likewise, scale invariance can also be achieved when the user provides the weights with the following expression:

$$\mathbf{x} = \arg\min_{\mathbf{x},\alpha} \|\overline{\mathbf{G}}\mathbf{x} - (\alpha\mathbf{f} - \mathbf{c})\| \tag{11}$$

Here, $\mathbf{f}$ denotes the resulting feature vector calculated with the given weights and α denotes the free scaling parameter. Though, to have a solution and to see the effect of scale invariance, at least two vertices must be constrained. Otherwise, it yields the trivial solution where the free scaling parameter is zero.

Shape Optimization

Current shape optimization techniques typically involve parameterized geometric models. Often times, this parameterization is not trivial and has to be established in advance by the designers. As an alternative to this practice, we propose using design spaces which inherently contain a parameterization model in terms of a geometric topology whose deformation characteristics and limits are controlled by the shapes and contributions of constituent exemplars, respectively. The parameterization is automatically set during sketching by providing different sketches with consistent topologies. Moreover, since design spaces are defined over constituent sketches, the space is spanned by adjusting contributions of these exemplars. Additionally, design spaces automatically convey geometric design knowledge from constituent sketches to the resulting design without a specific mathematical parameterization. As a result, the burden of defining a problem specific mathematical parameterization model for each different design is eliminated.

As mentioned earlier, our system is invariant to scaling. Hence, the resulting shapes do not readily have a proper scale to be able to work with dimensioned geometric constraints. For scale adjustment, we introduce a free scaling parameter which is also controlled by the optimizer while it explores the design space by adjusting contributions of the exemplars. The design domain can be bounded by constraining the exemplar weights, the free scaling parameter and any other quantity that can be defined using the resulting geometry (*e.g.* the area of a closed shape can be upper and lower bounded).

We demonstrate the optimization capabilities on two industrial design applications. The first example is a weight minimization problem to illustrate shape optimization under predefined engineering constraints, while the second example illustrates shape optimization under geometric constraints for both engineering and ergonomics considerations. In all calculations, the Sequential Quadratic Programming implementation of MATLAB© is used.

EXAMPLES

Bottle Example

To demonstrate our system's capabilities in shape optimization a bottle is designed. Three different bottle designs, namely Coke, wine and beer bottles, are sketched (Fig. 7.a) to construct a design space (Fig. 7.b). To illustrate manual design space exploration, two designs are synthesized with various weights including negative values, and resulting designs are given in Figure 7.c. It should be noted that using negative weights can also yield feasible design solutions.

Using the same design space, a bottle design having a desired liquid capacity with minimal weight is designed. The constraints on the bottle are given in Figure 7.d. Additionally, the sum of exemplar weights is kept unity while the upper and lower bound for weights are hand-picked to be 1.2 and -0.5, respectively. The bottom end of the contour is fixed at the origin to ensure that revolution of the contour around vertical axis at the origin results in a bottle. The upper end is fixed at constant radius for ergonomics considerations while its height is controlled by a scaling parameter. A linear thickness profile increasing towards the bottom is defined as a function of outer contour chord length. The parameters of the thickness profile are kept constant throughout the optimization. The bottle shape is optimized controlling the three exemplar weights and the scaling parameters, to a total of four design parameters. Once the resulting outer contour is calculated from the design space, the inner contour is calculated accordingly using the thickness profile. The weight and the inner volume of the bottle is then calculated after revolving the inner and outer contours around the vertical axis at origin to get the 3D bottle geometry.

The optimization process resulted in the design given in Figure 7.e with resulting weights, inner volume and neck radius. The optimum design resembles the beer bottle design more than the other exemplars. This is confirmed by the resulting exemplar weights. While the beer bottle design has the highest contribution, the coke bottle design has negative contribution. The contributions suggest that the beer bottle design promotes a high volume/weight ratio, whereas the Coke bottle design does not. However, by assigning a negative weight, the Coke bottle design is also used in favor of the objective.

Mouse Example

In this example, we demonstrate our method by designing a mouse under geometric constraints due to internal structures and an external human-device interface. A synthetic internal structure of a wireless mouse is given in Figure 8.d, showing circuit board, button bases, wheel and its base, and batteries. In Figure 8.e, a hand model in a relaxed position that can accommodate a mouse is shown. A mouse design which can encapsulate the internal structures while fitting into external human-device interface is sought. For this purpose, two design paces are defined for top and side views using three different mouse designs shown in Figure 8.(b,c). In both design spaces (*i.e.* both views) the rightmost points are constrained to be at the origin. New designs are calculated for both design spaces. The resulting shape is then scaled and positioned by using free variables. For consistency, the variable which controls the position of the mouse on axial direction is the same for both design spaces. Likewise, the same set of contributions of constituent exemplars is used for both design spaces. The optimizer controls three positioning, one scaling parameters and three exemplar contributions to a total of seven design parameters. The resulting shape is checked for interference with the internal and external constraints by calculating intersections of areas. The objective function to be minimized is defined as the total area of interference. Moreover, the sum of exemplar weights is kept unity while the upper and lower bounds for weights are hand-picked to be 1.2 and -0.2, respectively.

The resulting design satisfied both internal and external constraints as shown in Figure 8.f. The resulting shape has almost equal contributions from all three designs. Although first mouse design responded better to ergonomic constraints at both sides (*i.e.* thumb and little finger), contributions of other exemplars was required for satisfy all geometric constraints simultaneously. Although the constraints were applied on the outer contours for both views, the resulting shape also blends details, namely the button separation line and the character lines.

CONCLUSIONS AND FUTURE WORK

We described our method which assists early styling design of visually desirable industrial products through a sketch-based design interface. It provides means to initiate and develop design spaces from early conceptual sketches, and to explore new designs for a wide variety of industrial products. It assists designers to work on and modify a working geometry while it automatically presents feasible and meaningful solutions that are aligned with their modifications. It also permits exploration of design spaces by manually adjusting contributions of constituent sketches. By allowing flexible deformations, it can produce designs that are not limited to provided exemplars. Using design spaces permits integrated shape optimization under engineering and ergonomics constraints in early conceptual design. Thus, it unifies style and aesthetics considerations with engineering and

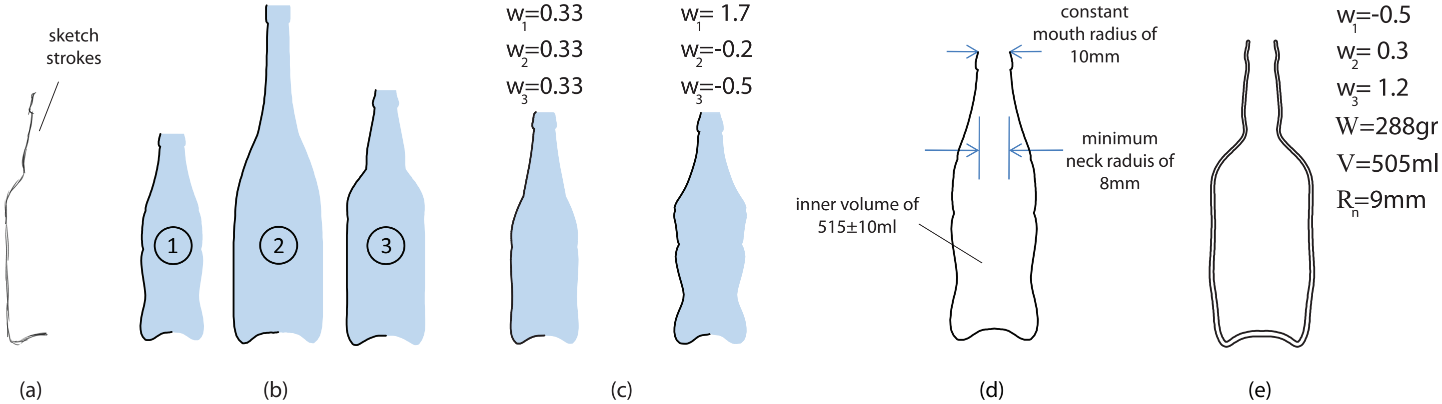

Figure 7. Design space is defined by (a) sketching (b) Coke, wine and beer bottles. The bottle is formed by revolving the drawn contours. The size variation among designs should be noted. (c) The design space is explored by manually setting exemplar weights. (d) Geometric constraints are imposed on bottle mouth, neck and inner volume. (e) The optimum design is given with resulting weights.

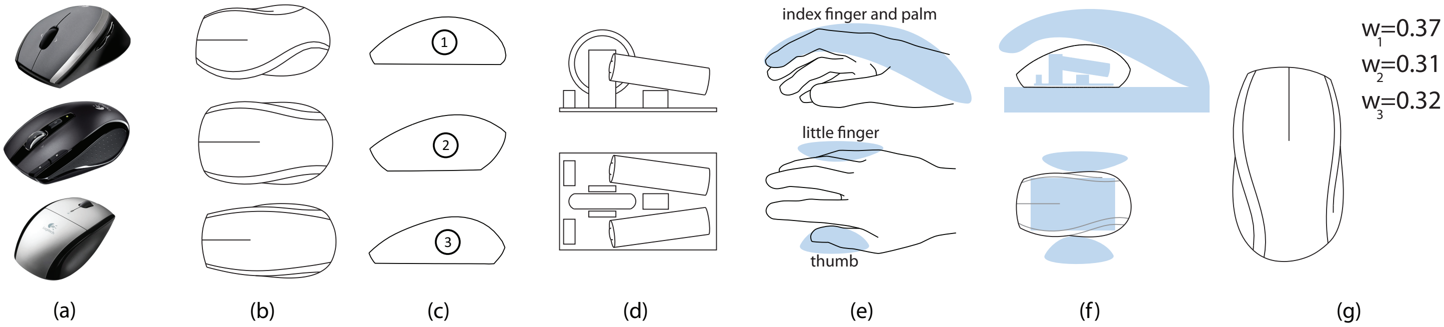

Figure 8. Mouse design: (a) Original mice models are used as templates to define two coupled design spaces; (b) one in top view (c) and one in side view. (d) Under internal (a synthetic internal structure including circuit board, wheel base, buttons, optic receiver and batteries is given) and (e) external constraints (human-device interface), (f,g) the optimum solution is determined.

ergonomics considerations early in design process. To do that, it does not require a predefined parameterization, instead it uses the contributions of constituent exemplars as design parameters of optimization. Therefore, it eliminates the need for specific mathematical models for different designs, instead the problem is automatically defined as designers draw their sketches. We demonstrated the capabilities of our method through examples on shape creation and optimization.

While our method presents effective means to assist early conceptual design activities, there are several avenues for further improvements. Currently our method requires one to one geometric correspondence between provided sketches. Although it detects possible joints in present graphics representation, designers are still required to provide consistent topologies among exemplars. We plan to expand our work to allow using different topologies in constructing design spaces. For this, we will explore automatic registration methods using graph matching techniques similar to those presented in [27]. Moreover, although optimization problems involving only geometric constraints are generalizable, it is still laborious to define other engineering optimization problems manually. Furthermore, currently our method supports only 2D sketches, yet with new sketch-based 3D modeling techniques it can be extended to create 3D design spaces. With this, our system can directly produce 3D geometries for product design involving geometric constraints due to internal structures and external human-device interfaces.

ACKNOWLEDGEMENTS

All mouse designs used in this work are courtesy of Logitech© Inc.

REFERENCES

[1] Karl T. Ulrich and Steven D. Eppinger. *Product Design and Development*. McGraw-Hill, Irwin, 4th edition edition,

2008.

[2] M. C. Yang. Concept generation and sketching: Correlations with design outcome. In *Proceedings of 2003 ASME Design Engineering Technical Conferences*, Chicago, IL, USA, Sept 2-6 2003.

[3] D. G. Ullman, S. Wood, and D. Craig. The importance of drawing in the mechanical design process. *Computers & Graphics*, 14(2):263–274, 1990.

[4] M. Schütze, P. Sachse, and A. Römer. Support value of sketching in the design process. *Research in Engineering Design*, 14(2):89–97, 2003.

[5] J. J. Shah, N. Vargas-Hernandez, J. D. Summers, and S. Kulkarni. Collaborative sketching (c-sketch) - an idea generation technique for engineering design. *The Journal of Creative Behavior*, 35(3):168–198, 2001.

[6] C. E. Catalano, B. Falcidieno, F. Giannini, and M. Monti. A survey of computer-aided modeling tools for aesthetic design. *Journal of Computing and Information Science in Engineering*, 2(1):11–20, 2002.

[7] S. N. Smyth and D. R. Wallace. Towards the synthesis of aesthetic product form. In *ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Proceedings of DETC'00, pages 218–224, Baltimore, Maryland, 2000.

[8] J. H. Frazer, J. M. Frazer, X. Liu, M. X. Tang, and P. Janssen. Generative and evolutionary techniques for building envelope design. In *Proceedings of 5th International Conference on Generative Art*, Milan, 2002.

[9] B. G. Bezirtzis, M. Lewis, and C. Christeson. Interactive evolution for industrial design. In *Proceedings of the 6th ACM SIGCHI Conference on Creativity & Cognition*, Washington, DC, USA, 2007. ACM.

[10] S. Wannarumon, E. L. J. Bohez, and K. Annanon. Aesthetic evolutionary algorithm for fractal-based user-centered jewelry design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 22(01):19–39, 2007.

[11] P. J. Bentley and J. P. Wakefield. Conceptual evolutionary design by genetic algorithms. *Engineering Design and Automation Journal*, 3(2):119–131, 1997.

[12] J. Cagan and M. Argawal. A bland of different tastes: the language of coffeemakers. *Environment and Planning B: Planning and Design*, 25:205–226, 1998.

[13] J. P. McCormack, J. Cagan, and C. M. Vogel. Speaking the buick language: Capturing, understanding, and exploring brand identity with shape grammars. *Design Studies*, 25:1–29, 2004.

[14] S. Orsborn, J. Cagan, R. Pawlicki, and R. C. Smith. Creating cross-over vehicles: Defining and combining vehicle classes using shape grammars. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 20:217–246, 2006.

[15] C. S. Randall, P. Richard, I. Kókai, J. Finger, and T. Vetter. Navigating in a shape space of registered models. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1552–1559, 2007.

[16] I. Kókai, J. Finger, C. S. Randall, P. Richard, and V. Thomas. Example-based conceptual styling framework for automotive shapes. In *Proceedings of the 4th EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*, Riverside, California, 2007. ACM.

[17] J. Moss, J. Cagan, and K. Kotovsky. Learning from design experience in an agent-based design system. *Research in Engineering Design*, 15:77–92, 2004.

[18] T. Kurtoglu and M. I. Campbell. An evaluation scheme of assessing the worth of automatically generated design alternatives. *Journal of Research in Engineering Design, In Press*, 2009.

[19] R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. Sketch: An interface for sketching 3d scenes. In *SIGGRAPH 96 Conference Proceeding*, pages 163–170, August 1996.

[20] T. Igarashi and J. F. Hughes. A suggestive interface for 3d drawing. *Proceedings of ACM Symposium on User Interface Software and Technology, UIST'01*, 2001.

[21] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3d freefrom design. In *ACM SIGGRAPH 99*, Los Angeles, California, August 1999.

[22] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Fibermesh: Designing freeform surfaces with 3d curves. In *ACM SIGGRAPH 2007*, ACM Transactions on Computer Graphics, San Diego, USA, 2007.

[23] O. A. Karpenko and J. F. Hughes. Smoothsketch: 3d freeform shapes from complex sketches. *ACM Transactions on Graphics*, 25/3:589–598, 2006.

[24] L. B. Kara and K. Shimada. Sketch-based 3d-shape creation for industrial styling design. *IEEE Computer Graphics and Applications*, 27(1):60–71, 2007.

[25] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popovic. Mesh-based inverse kinematics. *ACM Transactions on Graphics*, 24(3):488, 2005.

[26] K. Shoemake and T. Duff. Matrix animation and polar decomposition. In *Proceeding of Graphics Interface 92*, pages 259–264, 1992.

[27] W. Lee, L. B. Kara, and T. F. Stahovich. An image-based, trainable symbol recognizer for hand-drawn sketches. *Computers & Graphics*, 31(4):554–567, 2007.