PREDICTIVE MODELING FOR 2D FORM DESIGN

Erhan Batuhan Arisoy, Gunay Orbay, and Levent Burak Kara *

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
{earisoy,gorbay,lkara}@andrew.cmu.edu

Abstract. *In product design, designers often generate a large number of concepts in the form of sketches and drawings to develop and communicate their ideas. Concrete concepts typically evolve through a progressive refinement of initially coarse and ambiguous ideas. However, a lack of suitable means to visualize the emerging form at these early stages forces the designer to constantly maintain and negotiate an elusive mental image. To assist this process, we describe a predictive modeling technique that allows early, incomplete 2D sketches to be transformed into suggestive complete models. This helps designers take a sneak peek at the potential end result of a developing concept, without forcing them to commit to the suggestion. We demonstrate and discuss preliminary results of our technique on 2D shape design problems.*

Keywords: CAD, predictive modeling, conceptual design

1 Introduction

Recent developments in computer-aided design (CAD) have led to a large number professional software packages for different modeling needs. These packages all share a common goal, that is, to enable design of sophisticated geometries with less effort and prior experience. Nevertheless, most of these professional tools still require acquaintance with the underlying geometrical representations and related modeling operations and lack of suitable means to quickly visualize the emerging design ideas forces the designer to constantly maintain the mental image without any visual reference. To enhance this process, we propose a predictive form modeling method that converts early, incomplete drawings into suggestive complete models to establish visual references.

The main challenge in the proposed work is being able to interpret incomplete sketches to find a complete geometric model biased toward simpler interpretations, aligned with Occam's Razor theorem [12]. Our proposed technique has following specific contributions:

* lkara@andrew.cmu.edu Address all correspondence to this author.

1. Generating a suggestive prediction for the intended geometry from user drawn strokes
2. Enabling early visualization of the intended geometry for rapid evaluation and refinement decisions
3. Creating a feedback mechanism to check feasibility of the developed concept under proposed constraints

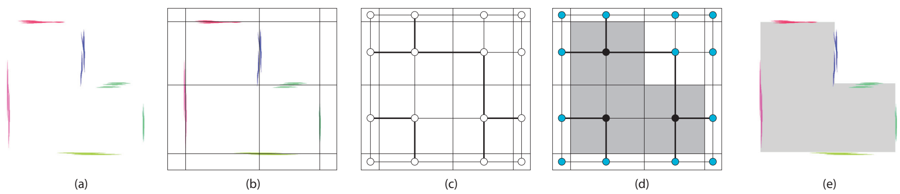

Fig. 1. The inner workings of our preliminary predictive drawing tool. (a) The input is a set of strokes that are grouped as they are drawn by the user. (b) The strokes are converted into geometric entities that partition the drawing plane. (c) The resulting partitions are represented as a graph structure. (d) The graph structure allows calculation of a region that is blocked by the input strokes (e) and visualized for the user.

In a typical scenario, our method takes as input a set of 2D sketches with 2D coordinate values and performs stroke clustering and beautification. In the next step, each stroke is treated as an infinitely long line which partitions the drawing plane into two regions and our method seeks to identify a union of these regions generated by every stroke as the prediction for the intended geometry through a genetic algorithm (GA).

2 Related Work

Researchers have extensively studied line drawings and their utilization in a wide variety of application areas ranging from non-photorealistic rendering techniques to 3D reconstruction. However, these line drawings typically involve roughly drawn overlapping strokes that require a separate analysis to identify salient curves and regions. To this end, Grabli *et al.* [2] and Wilson *et al.* [3] proposed line omission techniques where original strokes with less effectiveness scores are eliminated from the input set. Similarly, Barla and Sheh [4] developed simplification schemes based on perceptual and spatial grouping. In addition to sketch beautification and simplification, Sezgin [6] and Fu [7] focused on recognition and utilization of 2D user drawings and proposed tools designed to detect symbol networks from user drawings and convert them into engineering diagrams. Similarly, researchers teased out the importance of drawing order in sketches and Tversky and Suwa presented that the drawing order gives clues to the mental organization of the human brain [1]. Likewise, Novik and Tversky[8] supported this argument such that the drawing order reflects how the designers construct a concept in his mind. In addition to these studies, a multitude of tools are proposed for rapid 3D reconstruction using rough 2D sketches.Zeleznik [9] proposed a modeling tool which seeks to fit 3D primitives by grouping user drawn strokes. Igarashi [10, 5] utilized user drawn strokes for initial model generation and its deformation. The main advantage of the proposed system is bringing a higher level of prediction capability such that the prediction depends not only on one to one relationships but also on the entire set of user drawn strokes.

3 Technical Details

3.1 Overview

In this work, we develop and utilize a novel prediction approach that allows early, incomplete sketches to be transformed into suggestive geometries and our approach consists of 3 main steps: (1) Generation of input curve set, its simplification and beautification, (2) Construction of an undirected graph structure and (3) Definition of an objective function and its minimization on the constructed graph and visualization of the results.

3.2 Preprocessing of the Input Strokes

In this step, an automatic stroke grouping algorithm groups and separates the input strokes into distinct clusters of curves and each stroke is analyzed using two geometric pairwise stroke features. The first feature is the closest distance between the two strokes, and the second is the angle difference between the tangents that are drawn at the points that yield the closest distance between the strokes. After grouping, each stroke group is further analyzed and every stroke in the sketch are represented as line segments, arcs or circles.

3.3 Construction of Graph Structure

In our formulation, every user-drawn stroke is treated as an infinite line which partitions input domain into two different regions where our algorithm will assign either a positive or negative label for each region where positive assignments represent inclusions to the domain and negative assignments mean subtractions from the domain. We assign a probability on each extended line E_i such that they have a maximum value of 1 at the center position (C_x, C_y) of the representing stroke S_i and decreases as we move on E_i. Actually, we place a Gaussian distribution centered at (C_x, C_y) with mean 0 and variance depending on the length of S_i using equation (1). Figure 2 demonstrates the extension of beautifed user drawn strokes with corresponding probability assignments.

$$N(x) = e^{-(x-\mu)^2/2\sigma^2} where\, \sigma = \frac{\|E_i\|}{5}\, and\, \mu = 0 \tag{1}$$

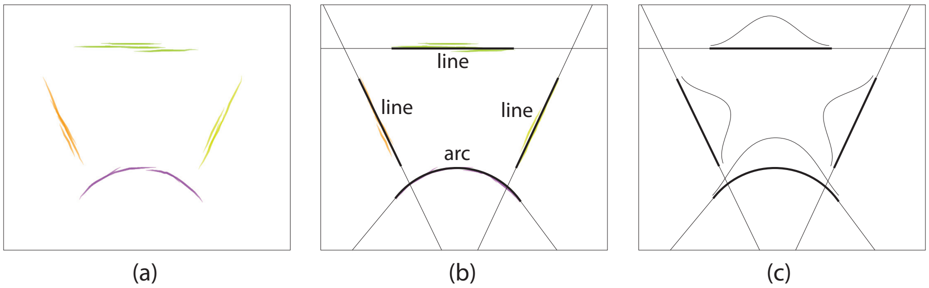

Fig. 2. The extension of the geometric entities to partition the drawing plane. (a) A given sketch, is first converted into a set of line or arc segments. (b) The lines and arcs are then extended at their end points in the tangent directions. (c) The extended lines attain weights according to a Gaussian kernel defined as a function of the segments length (*i.e.* influence).

Graph Structure Our graph model $G(V, E_d)$ consists of nodes N and undirected edges E_d connecting nodes to each other. In our problem setting, each node represents a partition created after the infinite extension of a stroke S_i in the image domain and each edge acts as the boundary existing between two partitions. Figure 3 demonstrates the conversion from partitioned image domain to the graph structure G. After creation of complete graph G, our algorithm calculates a probability for each edge based on the probability weights assigned in the stroke extension phase. The probability values exist on the extended lines enable us to have a metric to measure how likely the boundary between two partitions is a contour or not. This value is calculated by taking the mean of the corresponding probability values assigned on that portion of the extended line.

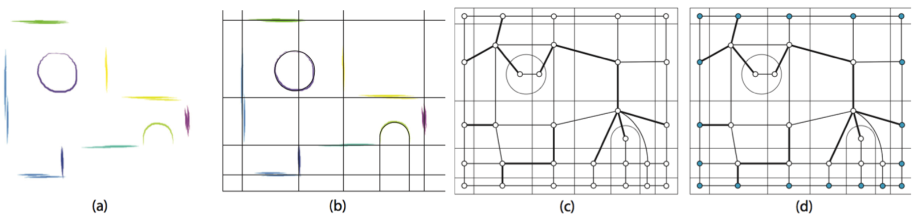

Fig. 3. The conversion of a sketch into a graph and the initial value assignments. (a) A given sketch is first converted into regions (b) by extending the identified lines, arcs, and circles. (c) Each resulting graph is represented as a node in the graph, while each adjacent region forms a link between their associated nodes. Here, the links that correspond to region boundaries that coincide with the original segments, are labeled as *constrained links*. (d) Initially, the regions that touch the drawing plane boundary are labeled as empty space.

3.4 Optimization and Objective Function

Our approach seeks to find the right combination of region labels through an optimization procedure where our objective function in Eqn. 2 consists of two main terms. First term, called assumed probability for an edge, represents the likelihood of an edge to be a part of the user drawn stroke intended for contour generation and second term is calculated based on the initial label assignments.

$$objectivefunction = \sum_{k=1}^{m} \sum_{i=1}^{n_k} (assumedweight_{ki} - realweight_{ki}) \qquad (2)$$

where m is the total number of nodes (i.e. number of partitions) and n_k is the number of first ring neighbors of the k^{th} node. Since our candidate solutions can be easily represented using chromosomes, we employ genetic algorithm to minimize our objective function using five steps explained in [11].

3.5 Results and Implementation Details

The interface of our algorithm is implemented in C++ and the optimization part is performed in MATLAB environment. Curve sets are generated using a graphics tablet interface. In all cases the domain is quantized into an $N \times N$ uniform grid. An important user adjustment parameter is the variance of the Gaussian distribution used in probability assignments on infinite lines. Users might prefer higher covariance values which increases the effect of the user drawn strokes or vice versa. The motivation is if the covariance is chosen very high, even a very short user drawn line will be treated as a very important contour separating inside form from the outside. The designer can modify this parameter according to his drawing style. Figure 4 illustrates two examples generated using proposed methodology where the domain size is 600 by 600 and covariance parameters are selected as 10. The optimization converged in 5 steps under 2 seconds. In terms of technicality, our main technical challenge is the fast increasing number of partitions with the introduction of each additional stroke to the domain and since each added partition expands dimension of the solution space and slows down the optimization. However, for most of the test cases, our GA method reaches global minimum in less than 50 iterations under 10 seconds. Furthermore, image based representation prevents us from dealing small area partitions and detection of neighborhood information using pixel data.

3.6 Conclusion and Future Work

In this paper we proposed a computer tool for early design stages which allows designers to take a sneak peek at the potential end results of an evolving design form. Our approach has been different than previous studies such that it can handle with both incomplete and higher genus geometries. We believe that future CAD tools will improve their auto complete or snap features to support more general prediction problems.

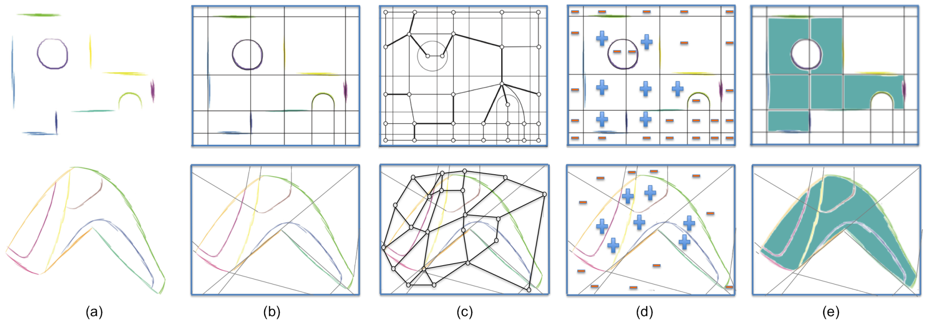

Fig. 4. Examples generated using proposed methodology. (a) 2D input drawings. (b) Extended lines and domain partitioning. (c) Corresponding graph structures. (d) Labels obtained via optimization. (e) Predicted solid regions.

References

1. Tversky, B., and Suwa, M., 2009. ". Thinking with sketches". *Tools for Innovation,* ***1***(9), pp. 75–85.
2. Grabli, S., Durand, F., and Sillion, F., 2004. "Density measure for line-drawing simplification". In Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on, pp. 309 – 318.
3. Wilson, B., and Ma, K.-L., 2004. "Rendering complexity in computer-generated pen-and-ink illustrations". In Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering, NPAR '04, ACM, pp. 129–137.
4. Barla, P., Thollot, J., and Sillion, F. X., 2005. "Geometric clustering for line drawing simplification". In ACM SIGGRAPH 2005 Sketches, SIGGRAPH '05, ACM.
5. Nealen, A., Igarashi, T., Sorkine, O., and Alexa, M., 2007. "Fibermesh: designing freeform surfaces with 3D curves". *ACM Trans. Graph.,* ***26***(3), p. 41.
6. Sezgin, T. M., Stahovich, T., and Davis, R., 2006. "Sketch based interfaces: early processing for sketch understanding". In ACM SIGGRAPH 2006 Courses, SIGGRAPH '06, ACM.
7. Fu, L., and Kara, L. B., 2011. "Neural network-based symbol recognition using a few labeled samples". *Computers & Graphics,* ***35***(5), pp. 955 – 966.
8. Novick, L., and Tversky, B., 1987. "Cognitive constraints on ordering operations: The case of geometric analogies.". *Journal of Experimental Psychology: General,* ***116***(1), p. 50.
9. LaViola, J., and Zeleznik, R., 2004. "Mathpad2: A system for the creation and exploration of mathematical sketches". In Proceedings of SIGGRAPH, Vol. 23, pp. 432–440.
10. Igarashi, T., Matsuoka, S., and Tanaka, H., 1999. "Teddy: a sketching interface for 3D freeform design". In In Proceedings of SIGGRAPH 1999, ACM, p. 21.
11. Goldberg, D., 1989. *Genetic algorithms in search, optimization, and machine learning.* Addison-wesley.
12. Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K., 1987. "Occam's razor". *Information processing letters,* ***24***(6), pp. 377–380.