Pacific Graphics (2019)
C. Theobalt, J. Lee, and G. Wetzstein (Editors)

RegionSketch: Interactive and Rapid Creation of 3D Models with Rich Details

Shuai Liu[1], Fei Hou[†2], Aimin Hao[1] and Hong Qin[‡3]

[1] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
[2] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[3] Department of Computer Science, Stony Brook University

Abstract

In this paper, we articulate a new approach to interactive generation of 3D models with rich details by way of sketching sparse 2D strokes. Our novel method is a natural extension of Poisson vector graphics (PVG). We design new algorithms that distinguish themselves from other existing sketch-based design systems with three unique features: (1) A novel sketch metaphor to create freeform surface based on Poisson's equation, which is simple, intuitive, and free of ambiguity; (2) Convenient and flexible user interface that affords the user to add rich details to the surface with simple sketch input; and (3) Rapid model creation with sparse strokes, which enables novice users to enjoy the utilities of our system to create expected 3D models. We validate the proposed method through a large repository of interactively sketched examples. Our experiments and produced results confirm that our new method is a simple yet efficient design tool for modeling free-form shapes with simple and intuitive 2D sketches input.

CCS Concepts

• ***Computing methodologies*** → *Sketch-based Modeling;*

1. Introduction and Motivation

In recent years, sketch-based modeling has outperformed conventional 3D geometric design processes with multiple modeling advantages, including simple user interface, natural and intuitive input using simple geometric primitives (i.e., curve sketches), reduced learning/training burden, lower time costs, etc. However, due to the inherent ambiguity, existing sketch-based modeling methods require redundant stroke input and could only create simple models without rich details and holes.

To solve the above problems of the existing methods, we present a method that allows users to model a freeform surface with complicated details and high genus using a more convenient sketching interface. We propose to make use of the Poisson's equation to construct smooth surface with complicated details and arbitrary holes. In addition, we introduce a new sketch interface, which allows users to control the concave-convex variation of the model flexibility. To derive complete 3D models, we extend our approach to multi-view: the user can rotate to a more opportune view to adjust the shape of the surface, and the patches produced by different views can be joining into a more complex model (see Fig. 7). Our salient contributions are as follows:

1. We propose a novel sketch metaphor to create freeform surface based on Poisson's equation, which is simple, intuitive and without ambiguity.
2. Our sketch metaphor is adequate to create 3D model with complicated details by a few and intuitive curves without topological restrictions.
3. We design and implement a sketch based 3D modeling system with simple and clear human-computer interface.

The remaining of the paper is organized as follows: Section 2 reviews the prior works. Section 3 introduces our single-view modeling method, then we shall describe our multi-view framework in Section 4, followed by the experimental results and comparisons with the latest works in Section 5. Finally, Section 6 concludes our method and points out our future work and limitations.

2. Related Works

Classic Geometric Inference Methods. Building a 3D shape from a 2D sketch is a challenge problem because of its inherent ambiguity. Existing solutions oftentimes have to make use of additional information and constraints to remove the ambiguities. Such as [IIMT07, JC08, NISA07, BPCB08, OSJ11, SWSJ06, TZF04, KH06] make an assumption that the model is symmetric in the sketch plane to quickly build smooth low-frequency 3D shapes. Since the stick figures are too sparse to express the surface, some systems

† Corresponding author: houfei@ios.ac.cn
‡ Corresponding author: qin@cs.stonybrook.edu

DOI: 10.2312/pg.20191331

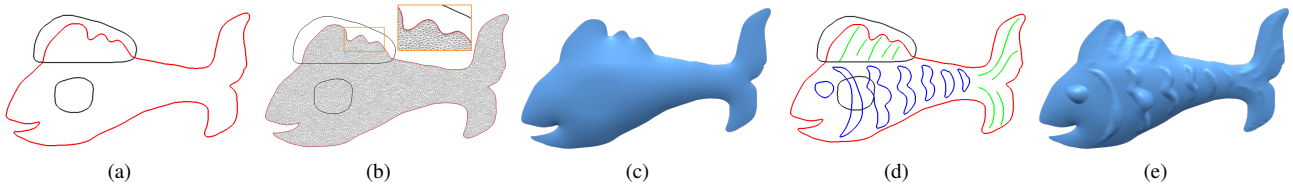


Figure 1: *Single-view modeling pipeline. (a) The input stroke primitives (red/black for boundary curve/soft bump curve), (b) The 2D domain mesh, (c) The basic surface, (d) The input stroke primitives (blue/green for hard bump region/feature curve), and (e) The final surface.*

[XGS15, BKL15, IBB15, LPL*17] make use of additional inputs to aid the modeling. In addition, geometric primitives and annotations are also commonly used in the field of sketch modeling [GIZ09, SAG*13, BCV*15]. But representing 3D shapes with geometric primitives often loses complicated details. Some previous works involve Poisson's equations to inflate meshes [SKC*14, FYXJ16, YHJ*17, JFZ*18, DNJ*18]. They used constant Laplacians to inflate regions with Neumann or Dirichlet boundary condition. However, this is not enough for sketch-based modeling. More recently, Hou et al. [HSF*18] presented a new vector primitive to control image shadings. Subsequently, Fu et al [FHS*18] extended these primitives on mesh decoration which supports level-of-details (LOD) editing on a closed 3D mesh. Inspired by their works, we extend these primitives to design an interactive sketch-based modeling system.

Data-driven Methods. The early data-driven based modeling methods generally search a shape from database against an input sketch and then deform the retrieved shape to fit the input sketch [LF08, XXM*13]. However, the results significantly depend on the database, and these methods are not capable for freeform shape modeling. Recently, the convolution neural network (CNN) is used to learn a regression model mapping sketches to 3D objects [NGDA*16, SDY*18, LGK*17, DAI*18]. The CNN can alleviate the difficulty of modeling but the CNN trained for a specific category cannot be generalized to handle other tasks. The latest work [LPL*18] uses a CNN to infer the depth and normal mapping for reconstructing surfaces from 2D sketches. However, CNN-based methods are difficult to precisely control the geometric details of the model.

3. Single-view Modeling

3.1. Curve Primitives

As show in the Fig. 1, we define the following 4 types of primitives to model the freeform surface:

- *boundary curve* - represents the boundary of the surface patch, and the curves are shown in red.
- *soft bump region* - adjusts the local bump of the surface with soft boundary, which is shown in black.
- *hard bump region* - adjusts the local bump of the surface with hard boundary, which is shown in blue.
- *line feature curve* - is used to produce a concave/convex line feature, which is drawn in green.

3.2. 2.5D Surface Generation

Planar triangulation. Our system automatically detects closed region Ω formed by the boundary curves and then we use the constrained Delaunay triangulation algorithm of CGAL [Yvi18] to quickly generate a dense mesh confined to the boundary curves (Fig. 1 (a) and (b)).

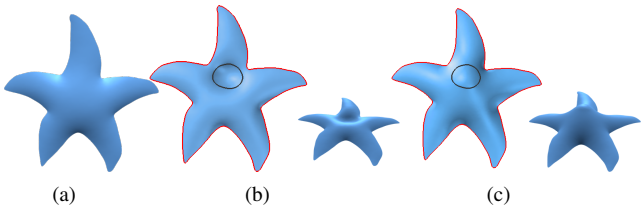


Figure 2: *Base model. (a) the smooth height field established on the region Ω, f is set to a constant ($f = 0.3$). (b) and (c) use soft bump region to divide the region Ω and assign different f_i values to different sub-regions ($f_{in} = -0.8$ for (b)) and ($f_{in} = 0.9$ for (c)), where f_{in} represents the Laplacian in the soft bump region (surrounded by black curve).*

Poisson's equation. Suppose the planar mesh M has n vertices, and we use a pair (E,V) to denote the mesh, where E is the set of edges and V is the set of vertices. We derive the surface by solving a Poisson's equation,

$$\begin{cases} \Delta\varphi(x) = f, & x \in \Omega\backslash\partial\Omega \\ \varphi\,|_{\partial\Omega} = g, & x \in \partial\Omega, \end{cases} \tag{1}$$

where Ω is the 2D domain enclosed the boundary curves sketched by user. $x \in \partial\Omega$ represents the boundary vertices of the triangular mesh M and $x \in \Omega\backslash\partial\Omega$ refers to the inner vertices of M. f is a function indicating the Laplacians of the surface. Consequently, the user can create a C^1 smooth surface model by setting different f values over different sub-regions through the intuitive sketch interface. We use $\varphi(x)$ to represent the solution, i.e., the 2.5D surface. The Dirichlet boundary condition g is determined by the boundary curves surrounding the region. We calculate the Poisson's equation by solving the following sparse linear system:

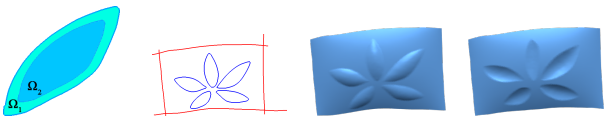
$$\mathbf{Lh} = \mathbf{b}, \tag{2}$$

where the matrix $\mathbf{L} \in \mathbf{R}^{\mathbf{n}\times\mathbf{n}}$ is the topological Laplacian of the mesh M [Sor05]. Given the Dirchlet boundary constraints, Eq. (2) becomes positive definite, so we solve it by Cholesky decomposition efficiently.

Soft bump region. As shown in Fig. 2, when the height field is constructed only by the boundary curves (Fig. 2(a)), the height propagates smoothly from the mesh boundary to the inner vertices. When the user draws a sub-region on the Ω using the soft bump region (Fig. 2 (b), (c)), the local concave-convex of the model can be easily adjusted through various values assigned to f in the the region. The concave/convex ambiguity is resolved by the sign of f. Notably, specifying different Laplacian values for sub-regions does not preserve the boundary shape clearly, and the high-frequency surface constructed by the boundary curve and the soft bump region primitives keeps C^1-continuity (Fig. 1 (c)).

3.3. Surface Decoration

Hard bump regions. Unlike the soft bump region, we design the hard bump region to create a bump with clear boundary. Inspired by [HSF*18], each hard bump region is determined by two sub-regions Ω_1 and Ω_2 (see Fig. 3(a)), and the outer sub-region Ω_1 should be very narrow to maintain the sharpness of the hard bump region. Then we specify a constant f_i for each of the two sub-regions of the hard bump region, where the signs of f_1 and f_2 are opposite, and we keep the integral value of the Laplacian in region Ω to 0 by adjusting the values of f_1 and f_2. Of course, our system hides these constraint details for the user, so the user only needs to drag a slider adjust the degree of bump of the geometric details. See Fig. 3 for an example.



(a) (b) (c) (d)

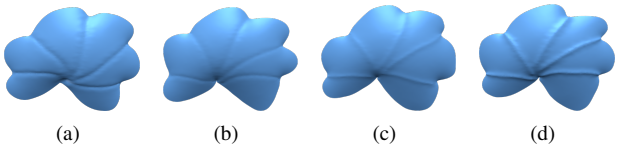
Figure 3: *Hard bump regions. (a) A hard bump region can be divided into two sub-regions, $\Omega = \Omega_1 \bigcup \Omega_2$. (b) Sketch input, where the red strokes are boundary curves and the blue strokes are hard bump regions. Different geometrical decorative effects can be produced by dragging the slider to adjust the degrees of the hard bump regions, where (c) $q = 0.5$ and (d) $q = -0.5$.*

Line feature curve. To create a narrow wrinkle (either concave or convex) on a surface, we design a line feature curve. When the user draws a line feature curve and sets the parameter p for it, our system first circles a narrow region of interest (ROI) along the primitive, and the ROI contains all the vertices within γ ($\gamma = 0.05$, about 4 times of the edge length of the triangle mesh). Second, our system updates the Laplacians of the vertices in the ROI, where the laplacians of the vertices on the line feature curve are f, the Laplacians of the internal vertices of the ROI are 0, and the Laplacians of the vertices of the ROI boundary is $-0.5f$. See Fig. 4 for an example.

4. Multi-view Modeling

Other auxiliary primitives. In the multi-view framework we will use the following two auxiliary curve primitives:

- *refine curve* -assists the users to adjust the height of the selected boundaries of the surface.

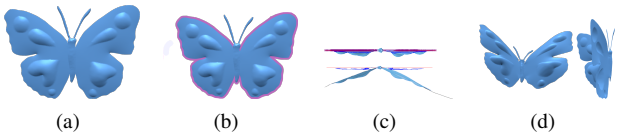


(a) (b) (c) (d)

Figure 4: *Various effects produced by line feature curves on the surfaces by adjusting the parameters. The parameters in (a) $p = -0.5$, (b) $p = -0.2$, (c) $p = 0.2$ and (d) $p = 0.5$.*

- *align curve* - is used to align the splicing boundaries of multiple patches.

Boundary curve height editing. To draw objects with complex boundary curves, the user can rotate the view and draw refine curve to edit the height of the selected boundary vertices at the new view (Fig. 5). Thanks to the solution characteristics of the Poisson equation, by modifying the boundary conditions in (Eq. (2)), the height of the boundary of our base surface can be easily modified.



(a) (b) (c) (d)

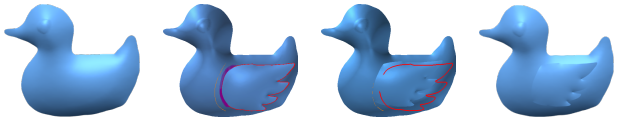
Figure 5: *Boundary curve height editing. (a) Butterfly model. (b) A purple curve to select the boundary vertices that need to adjust the height. (c) Adjust the model's boundary coordinates by drawing a refine curve from the top view. (d) After the deformed butterfly model, the geometric details on the butterfly's wings are reserved in the deformation.*

Multi-view sketching for 3D model generation. In order to solve the occlusion problem, we recommend that the user rotates the view angle to redraw the occluded surface. If for certain cases, the user wants the surface patch to be closely attached to the previous surface, we provide the align curve. As shown in Fig. 6, there is a gap between the duck wing drawn on the new drawing plane and the duck body (a). The user first selects the boundary vertices on the duck wrings and then draws an orange align curve on the duck body (b), then we use a shape-maintaining deformation method to eliminate the gap automatically (c)-(d).

Suppose there are m boundary vertices selected. We first obtain the differential coordinates δ_i [Sor05] of the vertices v_i. Then we update the position of the selected boundary vertices and solve the sparse linear system below:

$$\begin{bmatrix} \mathbf{L} \\ \mathbf{O}, \mathbf{I} \end{bmatrix} \mathbf{V} = \begin{bmatrix} \delta \\ \mathbf{v}' \end{bmatrix}, \tag{3}$$

where $\mathbf{L}$ is the topological Laplacian matrix of the mesh, $\mathbf{O}$ is a zero matrix, $\mathbf{I} \in \mathbf{R}^{\mathbf{m}\times\mathbf{m}}$ is the unit matrix, and $\mathbf{v}'$ is the new coordinates of the selected boundary vertices. Finally, we update the patch vertices by the solved $\mathbf{V} \in \mathbf{R}^{\mathbf{n}\times\mathbf{3}}$ matrix and eliminate the gap between the two patches.



(a) (b) (c) (d)

Figure 6: *Multi-view surface modeling. (a) A duck modeled from our single view modeling framework. (b) Adjust the view to create a new drawing plane to draw duck wing and then select the boundary vertices of the duck's wing to draw align curve (labeled in orange) on the duck's body. (c) After drawing align curve, the duck wings are tightly attached to the duck body. (d) Completed duck model.*

5. Results and Discussions

5.1. Experimental Results

We illustrate more free-form 3D surface models created by our system in Fig. 7, Fig. 8 and Fig. 10. These results demonstrate that our approach is simple yet powerful, and only a small number of stroke inputs can produce interesting models.

Rather than inflating local regions, we present a uniform framework to make the whole surface be the solution of a Poisson's equation. Our method can create models with high genus (Fig. 7 (a), (b), (c), (f) and (k)). We advance the use of Poisson's equation to generate 3D surface with different levels of details (Fig. 7 (c), (d), (g), (i), (k), (l) and (o)). These geometric details that are useful in 3D modeling are difficult to create with other existing sketch-based modeling tools. Besides, our system provides users with tools such as surface symmetry, sketch reuse, common view one-click switching, and auxiliary semi-transparent drawing boards to assist the user. In our experiments, an inexperienced user can skillfully use our sketch modeling system after several drawing attempts.

examples	vertices/faces	strokes (types)	meshing	height field	total
gecko (Fig. 10 (b))	5416/9413	16 (5)	58	83	663
leaf (Fig.7 (m))	10291/19314	16 (3)	94	288	1887
butterfly (Fig.5)	14371/27467	14 (3)	75	371	1623
cat mask (Fig.7 (c))	10682/20388	16 (3)	85	356	1375
Baymax (Fig.7 (e))	6443/12185	10 (2)	60	110	622
bag (Fig.7 (k))	6723/12787	12 (4)	70	116	869
butterfly (Fig.7 (a))	2213/4101	20 (1)	19	42	42

Table 1: *Running time statistics, in milliseconds. Where vertices/faces represent the number of vertices/faces of the model, strokes indicate the number of primitives in the sketch, and the types represent the number of primitive types used in the modeling. Meshing represents the time of triangulation, height field represents the time to construct the base surface, and the total is the sum of all time for 2D triangulation, height field calculation, height field modification and decorative geometry details, in ms.*

Performance. We validate our system on a PC with a 4 cores, 3.6GHz CPU. The running time of a typical sketch (Fig. 1) with 25 strokes and 9.2k faces takes about 82ms for 2D triangulation and 221ms to solve the Poisson's equation. And table 1 lists the time overhead of constructing different models by our system.

5.2. Comparisons

Bendsketch [LPL*17] defines a wide variety of curves types (see Fig. 8 (d) (e)), which are labeled in different colors. These complex stroke definitions increase the difficulty of user's using while it still cannot eliminate the bas-relief ambiguity. In contrast, we only provide four kinds of primitives with simple purposes and we use the sign of the parameter to eliminate the bas-relief ambiguity conveniently. As shown in the Fig 8, when constructing similar models, the type and number of 2D strokes that our system need are significantly less than these of [LPL*17]. What's more, [LPL*17] obtains the surface by solving an nonlinear optimization function. Therefore, it is obviously much more efficient to get the smooth surface by discretely solving the Poisson's equation on the mesh.

Subsequently, Li et al. [LPL*18] proposed a CNN network for the free-form surface modeling. Due to the data-driven approach, their predicted normal vectors of the CNN output contain noise, so it is necessary to extract the finial surface by Screened Poisson Reconstruction [KH13]. Considering that the Poisson surface reconstruction can only create watertight surfaces, these reconstructed surfaces cannot keep the high genus. Fig.9 (a) shows a failure case for [LPL*18]. (b) In contrast, the height field we calculated in the complex connected region can construct a smooth surface with holes directly. And ambiguity is a pervasive issue in data-driven and machine learning tasks. Therefore, their method is difficult to achieve precise control of geometric details.

6. Conclusion

In this article, we had proposed to employ the Poisson's equation to construct a height field representing the C^1 continuous smooth surface, and then we used two kinds of primitives: hard bump region and line feature curve to solve the geometrical decoration challenge. Finally, we made use of a multi-view modeling framework to construct more complex models and solve the self-occlusion problem of 3D models. We had verified the efficiency of the system through various experiments and comparisons. Compared with existing methods, our system could create freeform surfaces with rich geometric details and arbitrary typologies from simple and intuitive sketches.

Limitations and future work. The base patch created by our system must be a reflection of the height field. Even if we introduce the multi-view framework to solve the self-occlusion problem, creating models with more complex topology such as the torus knot, the braid structure object etc. is still a challenge for us. Those kind of more complex occlusion problems may be solved by adding hierarchical information to the stroke primitive, which we leave for future work. Our system performs well when creating smooth surface models with rich geometric details. However, we do not recommend that users use our system to create a complex structural model with a large number of planes, such as buildings or furniture. Because in our multi-view framework, the user needs to adjust the drawing panel manually and creating such models requires a series of complex camera operations.

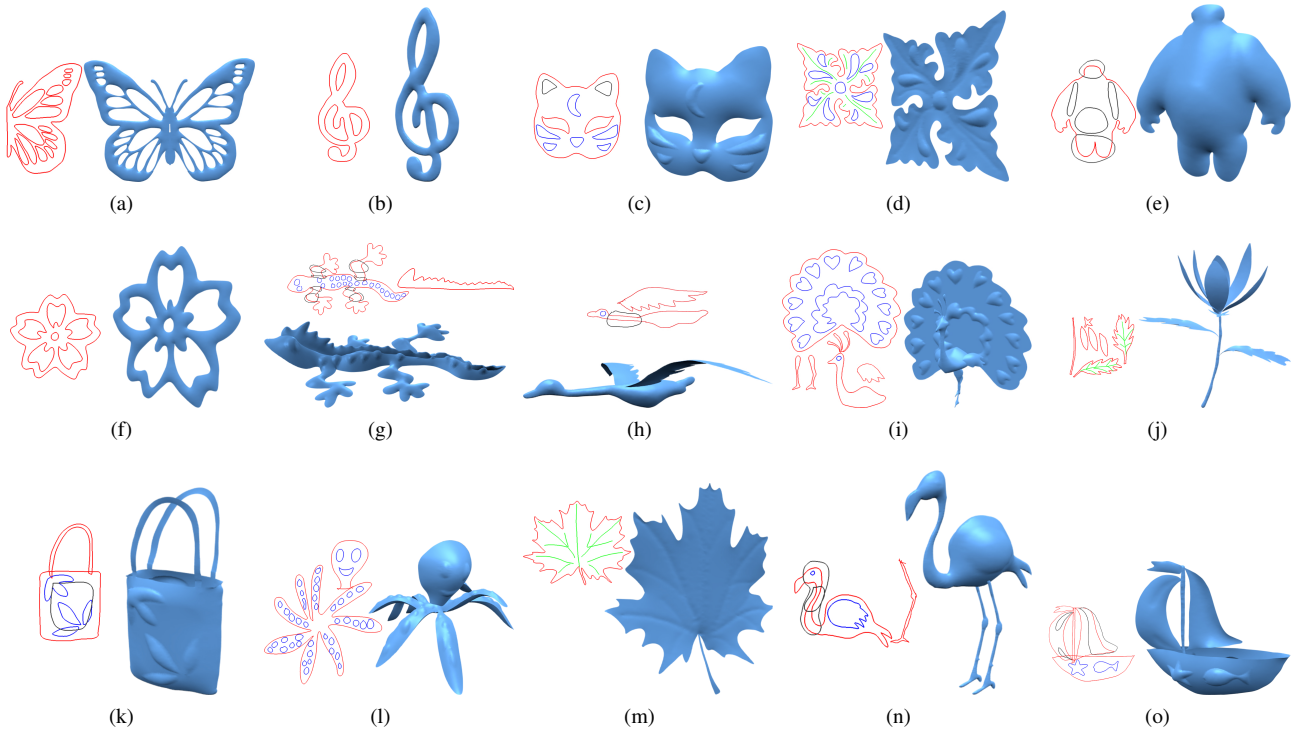


Figure 7: *3D models and their sketch inputs. The first six models ((a)-(f)) are generated by single-view modeling and the rest models are created by the multi-view modeling approach. Our four primitives: boundary curve, soft bump region, hard bump region and line feature curve are marked in the sketches in red, black, blue, and green, respectively.*

ACKNOWLEDGMENT

This work was partially supported by NSFC Grants (61872347, 61532002, 61672077), Special Plan for the Development of Distinguished Young Scientists of ISCAS (Y8RC535018), USA National Science Foundation IIS-1715985, IIS-1812606.

References

[BCV*15] BESSMELTSEV M., CHANG W., VINING N., SHEFFER A., SINGH K.: Modeling character canvases from cartoon drawings. *ACM Transactions on Graphics (TOG) 34*, 5 (2015), 162. 2

[BKL15] BUI M. T., KIM J., LEE Y.: 3d-look shading from contours and hatching strokes. *Comput. Graph. 51*, C (2015), 167–176. 2

[BPCB08] BERNHARDT A., PIHUIT A., CANI M.-P., BARTHE L.: Matisse: Painting 2d regions for modeling free-form shapes. In *SBM'08-Eurographics Workshop on Sketch-Based Interfaces and Modeling* (2008), Eurographics Association, pp. 57–64. 1

[DAI*18] DELANOY J., AUBRY M., ISOLA P., EFROS A., BOUSSEAU A.: 3d sketching using multi-view deep volumetric prediction. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 1*, 21 (2018). 2

[DNJ*18] DVOROÅ¿ÅĹAK M., NEJAD S. S., JAMRISKA O., JACOBSON A., KAVAN L., SÃ¡KORA D.: Seamless reconstruction of part-based high-relief models from hand-drawn images. 5. 2

[FHS*18] FU Q., HOU F., SUN Q., XIN S.-Q., LIU Y.-J., WANG W., QIN H., HE Y.: Decorating 3d models with poisson vector graphics. *Computer-Aided Design 102* (2018), 1–11. 2

[FYXJ16] FENG L., YANG X., XIAO S., JIANG F.: An interactive 2d-to-3d cartoon modeling system. 193–204. 2

[GIZ09] GINGOLD Y., IGARASHI T., ZORIN D.: Structured annotations for 2d-to-3d modeling. *ACM Transactions on Graphics (TOG) 28*, 5 (2009), 148:1–148:9. 2

[HSF*18] HOU F., SUN Q., FANG Z., LIU Y.-J., HU S.-M., HAO A., QIN H., HE Y.: Poisson vector graphics (pvg). *IEEE Transactions on Visualization and Computer Graphics* (2018), 1–12. 2, 3

[IBB15] IARUSSI E., BOMMES D., BOUSSEAU A.: Bendfields: Regularized curvature fields from rough concept sketches. *ACM Trans. Graph. 34*, 3 (2015), 24:1–24:16. 2

[IIMT07] IGARASHI T., IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3d freeform design. In *Acm siggraph 2007 courses* (2007), ACM, p. 21. 1

[JC08] JOSHI P., CARR N. A.: Repoussé: Automatic inflation of 2d artwork. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling* (2008), Eurographics Association, pp. 49–55. 1

[JFZ*18] JAYARAMAN P. K., FU C., ZHENG J., LIU X., WONG T.: Globally consistent wrinkle-aware shading of line drawings. *IEEE Transactions on Visualization and Computer Graphics 24*, 7 (2018), 2103–2117. 2

[KH06] KARPENKO O. A., HUGHES J. F.: Smoothsketch: 3d free-form shapes from complex sketches. In *ACM Transactions on Graphics (TOG)* (2006), vol. 25, ACM, pp. 589–598. 1

[KH13] KAZHDAN M. M., HOPPE H.: Screened poisson surface reconstruction. *ACM Transactions on Graphics 32*, 3 (2013), 29. 4

[LF08] LEE J., FUNKHOUSER T.: Sketch-based search and composi-

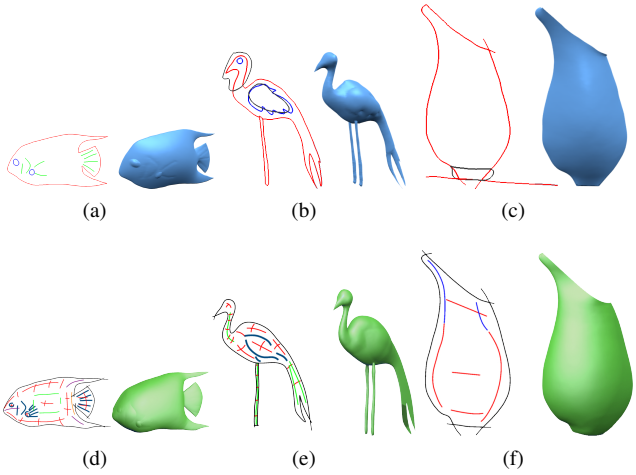


Figure 8: *Comparison with [LPL*17]. Ours models are in blue and theirs are in green. The left of every model illustrates the input sketch. Their method uses a large number of strokes with different types of labels (for example, line feature, ridge, valley and flat etc.. Each label is in a different stroke color). In contrast, our strokes are much simpler and more natural with a fewer types of stroke primitives.*

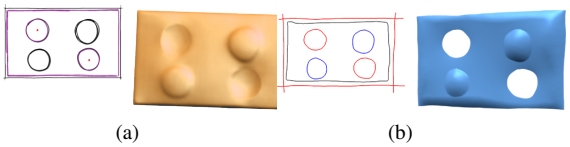


Figure 9: *A failure case for [LPL*18] is easily created by our method (Their model is in orange (a) and ours is in blue (b)). This is a bumpy plane that should contain two bumps and two holes but [LPL*18] can only create watertight surface.*

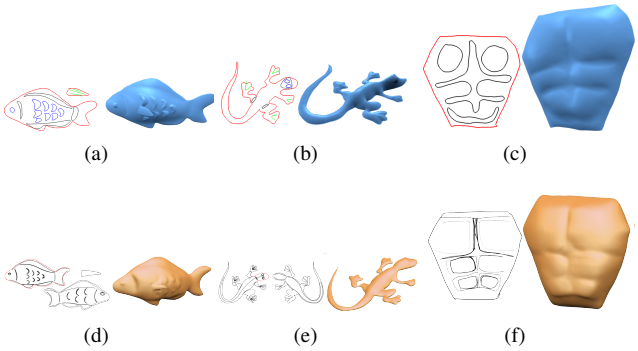


Figure 10: *Comparison with [LPL*18]. Ours models are in blue and theirs are in orange. The left of every model illustrates the input sketch. The feet of the gecko model in (e) loss the geometric details of the user input but our gecko feet (b) are not. In order to produce denser geometric details on the surfaces (d) (f), the sketches contain many overlapping detail strokes, but the details of the models are still blurred and rough. In contrast, our inputs are more concise and the resulting models have clearer geometric details (a) (c).*

tion of 3d models. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling* (2008), Eurographics Association, pp. 97–104. 2

[LGK*17] LUN Z., GADELHA M., KALOGERAKIS E., MAJI S., WANG R.: 3d shape reconstruction from sketches via multi-view convolutional networks. In *3D Vision (3DV), 2017 International Conference on* (2017), IEEE, pp. 67–77. 2

[LPL*17] LI C., PAN H., LIU Y., SHEFFER A., WANG W.: Bendsketch: Modeling freeform surfaces through 2d sketching. *ACM Trans. Graph. (SIGGRAPH) 36*, 4 (2017), 125:1–125:14. 2, 4, 6

[LPL*18] LI C., PAN H., LIU Y., SHEFFER A., WANG W.: Robust flow-guided neural prediction for sketch-based freeform surface modeling. *ACM Trans. Graph. (SIGGRAPH ASIA) 37*, 6 (2018), 238:1–238:12. 2, 4, 6

[NGDA*16] NISHIDA G., GARCIA-DORADO I., ALIAGA D. G., BENES B., BOUSSEAU A.: Interactive sketching of urban procedural models. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 130:1–130:11. 2

[NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Fibermesh: designing freeform surfaces with 3d curves. *ACM transactions on graphics (TOG) 26*, 3 (2007), 1–9. 1

[OSJ11] OLSEN L., SAMAVATI F., JORGE J.: Naturasketch: Modeling from images and natural sketches. *IEEE Computer Graphics and Applications 31*, 6 (2011), 24–34. 1

[SAG*13] SHTOF A., AGATHOS A., GINGOLD Y., SHAMIR A., COHEN-OR D.: Geosemantic snapping for sketch-based modeling. *Computer Graphics Forum 32*, 2 (2013), 245–253. 2

[SDY*18] SU W., DU D., YANG X., ZHOU S., FU H.: Interactive sketch-based normal map generation with deep neural networks. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 1* (07 2018), 1–17. 2

[SKC*14] SÃ¡KORA D., KAVAN L., CADIK M., JAMRISKA O., JACOBSON A., WHITED B., SIMMONS M., SORKINEHORNUNG O.: Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transactions on Graphics 33*, 2 (2014), 16. 2

[Sor05] SORKINE O.: Laplacian mesh processing. In *Eurographics 2005 - State of the Art Reports* (2005), Chrysanthou Y., Magnor M., (Eds.), The Eurographics Association. 2, 3

[SWSJ06] SCHMIDT R., WYVILL B., SOUSA M. C., JORGE J. A.: Shapeshop: Sketch-based solid modeling with blobtrees. In *ACM SIGGRAPH 2006 Courses* (2006), ACM, p. 14. 1

[TZF04] TAI C.-L., ZHANG H., FONG J. C.-K.: Prototype modeling from sketched silhouettes based on convolution surfaces. In *Computer graphics forum* (2004), vol. 23, Wiley Online Library, pp. 71–83. 1

[XGS15] XU Q., GINGOLD Y., SINGH K.: Inverse toon shading: Interactive normal field modeling with isophotes. In *Proceedings of the workshop on Sketch-Based Interfaces and Modeling* (2015), Eurographics Association, pp. 15–25. 2

[XXM*13] XIE X., XU K., MITRA N. J., COHEN-OR D., GONG W., SU Q., CHEN B.: Sketch-to-design: Context-based part assembly. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 233–245. 2

[YHJ*17] YEH C. K., HUANG S. Y., JAYARAMAN P. K., FU C., LEE T.: Interactive high-relief reconstruction for organic and double-sided objects from a photo. *IEEE Transactions on Visualization and Computer Graphics 23*, 7 (2017), 1796–1808. 2

[Yvi18] YVINEC M.: 2D triangulation. In *CGAL User and Reference Manual*, 4.13 ed. CGAL Editorial Board, 2018. 2