Globally Consistent Wrinkle-Aware Shading of Line Drawings

Pradeep Kumar Jayaraman, Chi-Wing Fu, Jianmin Zheng, Xueting Liu and Tien-Tsin Wong

Abstract—Shading is a tedious process for artists involved in 2D cartoon and manga production given the volume of contents that the artists have to prepare regularly over tight schedule. While we can automate shading production with the presence of geometry, it is impractical for artists to model the geometry for every single drawing. In this work, we aim to automate shading generation by analyzing the local shapes, connections, and spatial arrangement of wrinkle strokes in a clean line drawing. By this, artists can focus more on the design rather than the tedious manual editing work, and experiment with different shading effects under different conditions. To achieve this, we have made three key technical contributions. First, we model five perceptual cues by exploring relevant psychological principles to estimate the local depth profile around strokes. Second, we formulate stroke interpretation as a global optimization model that simultaneously balances different interpretations suggested by the perceptual cues and minimizes the interpretation discrepancy. Lastly, we develop a wrinkle-aware inflation method to generate a height field for the surface to support the shading region computation. In particular, we enable the generation of two commonly-used shading styles: 3D-like soft shading and manga-style flat shading.

1 Introduction

Shading can dramatically change the atmosphere and style of a 2D drawing. Artists achieve this by manually locating the shading regions, followed by coloring them, or laying screentones, based on the desired visual style. While such a process can be automated with the presence of geometry, it is impractical for artists to spend time on modeling the geometry for every single drawing. Therefore, it would be desirable to infer certain geometry information directly from the artists' clean line drawing, automate the shading, and save the artists from the tedious shading process, so that they can focus on creative aspects.

Early methods for modeling the geometry of line drawings [1], [2], [3], [4], [5], [6] seem to be applicable for our need. However, these methods are designed mainly for line drawings of rigid polyhedral shapes in orthographic view. In our case, arbitrary 2D drawings generally depict freeform objects with no guarantee on the physical correctness.

Sketch-based modeling methods [7], [8], [9] may be usable, but they are too demanding for our shading purpose as they aim to create a complete 3D geometry, requiring much user annotation. Note that traditional 2D cartooning is a highly labor-intensive process with a very tight schedule. An approximate view-dependent proxy geometry would be sufficient to support the shading, as shown by Sýkora et al. [10]. However, their method considers mainly the region contour, while we need to consider interior strokes, which depict intra-region bulges and dents.

Besides, some methods aim to recover the normal field [12], [13], [14] or proxy geometry [15], [16] from line drawings. However, they are not very successful in analyzing the semantics of interior strokes, e.g., see the wrinkle strokes in Fig. 1. As the way that these methods interpret strokes is either user-driven, or handled locally without a global consideration, the inferred local geometry may not be consistent over the line drawing. In particular, we may have ambiguities in the local interpretation, see Fig. 2. Such crowded wrinkle strokes are not rare in conventional 2D line drawings, unfortunately.

In this work, our goal is to automate shading production on line drawings. Particularly, we consider both boundary and interior strokes, but focus mainly on the interior wrinkle strokes, which are feature lines purposely drawn to illustrate geometric features of the underlying object, e.g., ridges/valleys [17], [18], suggestive contours [19], as well as occluding contours (a.k.a. silhouettes) [20], [21], [22]. Each of these strokes suggests certain sparse cues about the local shape or depth profile of the object in the drawing. It has been studied by Cole et al. [23] that these line drawing definitions can explain almost all kinds of human drawn lines in drawings (81% in their dataset). Therefore, by analyzing their semantics, we can infer local perceptual information, and construct partial geometric information over the line drawing to support shading production. Such an analysis is generally lacking in existing works.

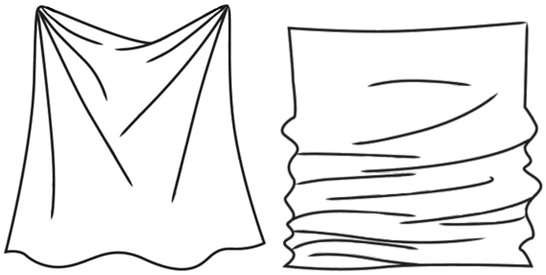


Fig. 1. With just a few strokes, line drawings [11] can illustrate nontrivial geometric features that are (perceptually) not difficult to recognize.

- *Pradeep Kumar Jayaraman and Jianmin Zheng are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.*
- *Chi-Wing Fu, Xueting Liu and Tien-Tsin Wong are with The Chinese University of Hong Kong, Hong Kong.*

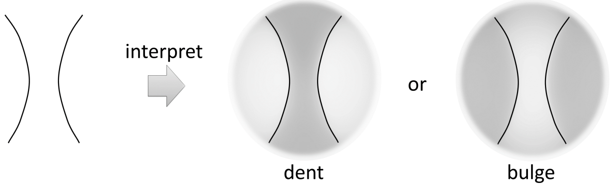


Fig. 2. Ambiguity in interpreting a pair of wrinkle strokes shown on the left. Locally, the in-between region can be perceived as a dent or a bulge; note: white and black colors indicate near and far, respectively.

To meet the goal, we develop a series of computational methods with the novel technical components below:

- First, we adopt and model five different perceptual cues, namely T-junction, convexity, continuity, proximity, and regularity, to interpret the local geometric meaning of wrinkle strokes by exploring relevant psychological principles. These cues not only suggest the local shape based on individual strokes, but also explore the spatial arrangement and interaction among strokes for consistent interpretation of strokes over the drawing.
- Second, we formulate a global optimization for the stroke interpretation problem, aiming at maximizing the fulfilment of local interpretations and minimizing the inconsistencies across adjacent interpretations.
- Lastly, we develop a wrinkle-aware inflation method to estimate the partial geometry of the line drawing for shading production. We support two commonly-used shading styles: 3D-like soft shading and flat shading in manga style (see Fig. 3c and Fig. 4 for examples).

To demonstrate the applicability and effectiveness of our method, we present shading results on variety of input line drawings, and conduct experiments to evaluate and compare our method against state-of-the-art methods.

2 Related Work

To shade a line drawing, one needs to first create or recover full 3D geometry or partial geometry (e.g., height field) of the drawing. For this purpose, there are generally two categories of methods: sketch-based modeling and image-based geometry reconstruction. Moreover, this work also relates to image-based shading and line interpretation.

Sketch-based Modeling. Sketch-based methods create 3D geometry from user sketches, e.g., Teddy [7], FiberMesh [8], NaturaSketch [24], True2Form [25], and SecondSkin [26]; see related surveys by Olsen et al. [27], and Cook et al. [28] for details. Usually these methods require users to repeatedly draw strokes or add depth clues to define or modify the geometry. E.g., FiberMesh uses extra control curves to cut, extrude, or tunnel a shape. This process enables the creation of nontrivial geometry, but is rather tedious and often more than necessary in our case, since we only need to produce shading without requiring a physically-correct geometry.

Estimating 3D Geometry from 2D Lines/Images. Extensive research has been done in reconstructing 3D or pseudo-3D geometry from a 2D natural image, particularly in recent years, e.g., [29], [30], [31], [32], to name a few. Rather than attempting to be exhaustive, we mainly focus on methods for 2D line drawings, since they are more related to this work. To construct a 3D geometry from a line drawing, various works have been developed for industrial and architectural line drawings with strong assumptions on the input drawings and shapes being constructed [33], [34]. Hence, these methods generally cannot handle arbitrary drawings. Existing geometry reconstruction methods to handle arbitrary line drawings can be roughly classified into inflation methods and normal field estimation methods.

Inflation methods reconstruct 3D geometry in an input drawing by blowing up each 2D object region to 3D. To create detailed structures in a region, users may specify certain internal constraints [15], [16], [32], [35], [36]. While these interactive methods produce good results given sufficient user-specified constraints, they require tedious manual inputs to handle crowded wrinkle strokes. Besides user-specified constraints, attempts have been made to automatically construct surface geometry from line drawings by analyzing T-junctions along boundary lines [37]. However, the analysis of interior strokes is generally lacking. Although some methods handle interior strokes by region subdivision [10], this approach requires user input and additional computation to ensure that the subdivided regions can be integrated in the final geometry. Kolomenkin et al. [38] reconstructed archaeological line drawings whose interior strokes represent flat step-like features with similar profiles; however, the method is unable to handle freeform drawings with wrinkled surface and complex local features. Some other works explored ways to reconstruct a developable surface from 3D curves [39], or a set of multi-view 2D drawings [40]. These methods cannot be applied to our case, since we only have a single 2D line drawing as input. Our method estimates partial local geometry by analyzing the semantics of strokes in the drawing, and then infers a delicate geometry via a global optimization. Hence, we can support shading production even for drawings with complex wrinkle strokes.

Normal field estimation methods reconstruct a normal field over the image space of the input drawing using various features, e.g., dominant curvatures [12], user-marked normals [41], cross-sectional curvature lines [13], [14], 3D normals along isophotes [42], and hatching strokes [43]. While most of these methods can well reconstruct the global shape of the drawing, they generally cannot handle the interior strokes that indicate wrinkles or folds. To this end, Johnston [12] incorporated manual specification to indicate the sign of normals along each stroke whenever needed, while Bui et al. [43] estimated local geometry along each stroke using nearby hatching strokes. In a short summary, existing methods adopt only simple local features (e.g., curvature) to estimate local geometry, and require additional manual specifications (e.g., label and hatching strokes) to achieve good results. In contrast, we analyze not only local stroke properties and their spatial interactions, but also formulate a global optimization to produce a global interpretation of strokes.

Shading 2D Drawings. Artists commonly create shading by imagining the 3D shape and light source location, and shading the associated regions with darker colors, gradients or textures, e.g., hatching and screening. While we may use software like Photoshop, Illustrator, and Manga Studio, the manual process is still tedious. Only a few works focused on

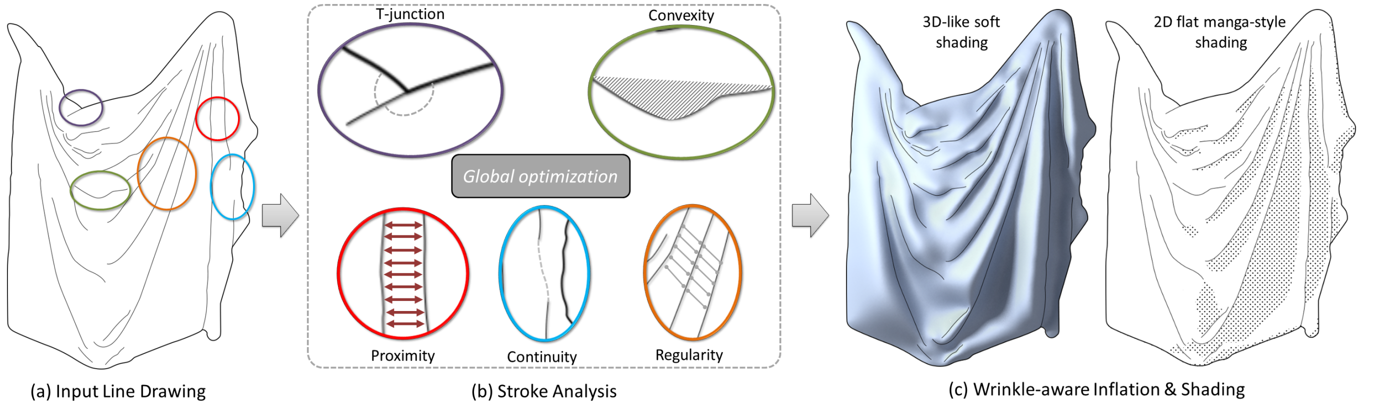


Fig. 3. Overview of our approach. We take a 2D line drawing (TOWEL) as input (a), and analyze various perceptual cues to estimate the local geometric information around each stroke (b). After that, we combine these local cues via an optimization model to maximize the satisfaction over the local inference and to obtain a globally-consistent stroke interpretation. Lastly, we develop a wrinkle-aware inflation method to generate a partial geometry, which can be used to support the production of two different shading styles (c).

simplifying the process, e.g., Qu et al. [44] proposed a manga screening method to convert color photos to manga images; however, since their method relies on color and texture information, it cannot handle line drawings. This work aims to take a clean line drawing as input, and estimate the shading regions to automate shading production.

Line Interpretation. The interpretation of lines or user-drawn strokes has gained interest from both perceptual studies and computer vision research. Feature lines [45], [46] are particularly considered, since they are purposely drawn [23] to depict geometric features such as silhouettes, sudden changes in gradient, folds/bends in the surface, etc. These lines provide important cues about the 3D shape that can be consistently interpreted by humans [47].

To interpret lines, we may explore local (individual lines) and global features (among lines). Local features like 3D surface properties such as principal curvatures [48], [49], [50] and depth/orientation discontinuities [2], [4] have been extensively studied from a geometric perspective. By exploring the Gestalt psychology [51], [52], higher-level semantics of lines could be estimated by considering human perception. Such results can facilitate a wide range of applications, e.g., curve completion [53], local layering [54], [55], [56], line drawing simplification [57], [58], etc. Compared to previous works, we also interpret lines as guided by the Gestalt psychology, but focus particularly on wrinkled surfaces illustrated in line drawings. Moreover, our method considers several perceptual cues and how strokes interact with one another to estimate partial geometry information through a global optimization.

3 Overview

Our Goal. Our input is a set of interior and boundary strokes in a given vectorized clean line drawing, e.g., see Fig. 3a. These strokes are purposely drawn by hand to illustrate various geometric features such as bulge, dent, and silhouette. Each stroke is simply given as a 2D polyline, without labels that indicate the geometric features.

Taking such input, our goal is to analyze the strokes in the given line drawing (particularly the wrinkle strokes), including their local shapes and how they spatially interact with one another, to estimate partial geometry information in the line drawing to automate shading production.

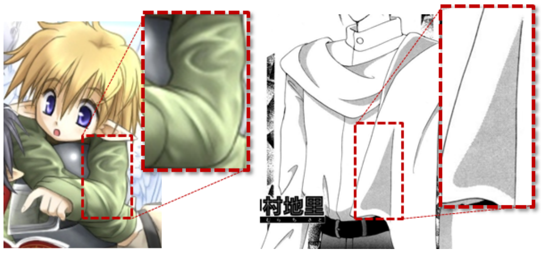


Fig. 4. Common shading styles. 3D-like soft shading (left) (by getty [60]) employs gradients and soft shadows, while manga-style flat shading (right) (e.g., from [61]) has hard shading region boundaries. Also note that the shading region seldom goes across strokes in flat shading.

The Challenges. To produce shading with wrinkles, we need to decide where to shade and how much to shade. To this end, we have to estimate certain local geometry information around the strokes, even though we do not require physical correctness. This is a challenging problem as our input is only a single line drawing without temporal information (in contrast to problems tackled by Zhang et al. [59] and Liu et al. [55]), and the strokes are simply 2D polylines without labels and user annotations. To resolve these issues, we consider the following subproblems:

- First, we have to estimate local geometry information around each stroke, specifically, i) the local gradient around the stroke, ii) specific geometric feature suggested by the stroke's shape or the presence of T-junction at the stroke endpoint(s), and iii) its interaction with nearby strokes. To support the analysis, T-junctions may help to estimate the local geometry [55], [56], but T-junctions are insufficient for resolving the problem, since most of them are located around the object boundary, while wrinkles in line drawings often occur in interior regions, see Fig. 1.
- Second, we need to integrate the local geometry information estimated from individual strokes, so that we can create a globally-consistent stroke interpretation and a partial geometry over the entire line drawing. If we only consider local cues around individual strokes, the estimated geometry may be ambiguous or featureless, see a comparison experiment in Sec. 6.4.
- Lastly, we need to consider the characteristics of shading

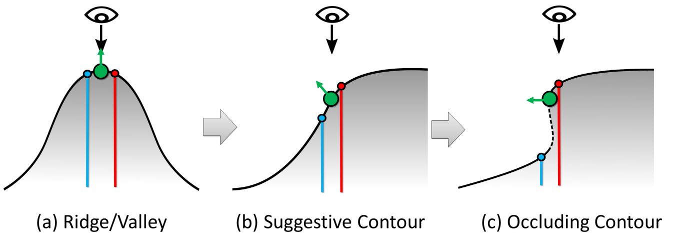


Fig. 5. These cross-sectional views illustrate transitions across depth profiles in the simplified stroke model, where the green dot refers to a point on the stroke. In this model, when the gradient value $|g_i|$ is small, the stroke is interpreted as a ridge or valley, and when $|g_i|$ increases, the local depth profile gradually transits to a suggestive contour, and further to an occluding contour with a local depth discontinuity.

styles. Two shading styles commonly found in cartoon and manga are considered (see Fig. 4 for examples):

i) *Soft shading* resembles 3D shading and is realized by means of color gradients. Such a style has a smooth transition from dark to light across a shading region and provides a nice 3D perception; however, since this style involves gradients, it is very tedious to create manually.

ii) *Flat shading* in manga style differs from conventional 3D shading; the shading region has hard boundary with solid color or texture, and usually stops at and *seldom* goes across the strokes in drawing.

Our Approach. We approach the above subproblems through a novel three-stage computation pipeline, see Fig. 3:

- First, we analyze not only T-junctions but also several other perceptual cues by consulting relevant principles from the Gestalt psychology. These include cues on individual strokes (convexity), and cues on pairs and groups of strokes (proximity, continuity, and regularity).
- Next, to obtain a globally-consistent stroke interpretation over the perceptual cues, we formulate a novel optimization model to combine and balance the local stroke inference results (Fig. 3b). Here, we define energy functions to quantify individual cues, and integrate them into a global energy function, so that we can maximize the perceptual consistency over the drawing.
- Lastly, to produce shading, we develop a wrinkle-aware inflation method to estimate a partial geometry by constructing a mesh and estimating its z-coordinates (a height field) from the local depth profiles inferred around strokes (Fig. 3c). We design our inflation framework such that the reconstructed mesh satisfies the characteristics of the associated shading style, i.e., 3D-like soft shading or flat shading in manga style.

4 Inferring Stroke Gradients

4.1 A Simplified Stroke Model for Wrinkles

We consider two kinds of strokes: i) for *boundary strokes* that enclose an object in the given drawing, we employ conventional boundary conditions to model their geometric behavior; see Sec. 5 for detail, and ii) for *interior strokes* (each denoted as s_i), we devise a simplified stroke model to describe their depth profiles. Like freeform 2D drawings, this model does not aim for physical correctness but for the estimation of perceptual information, and the construction of partial geometry to support shading production.

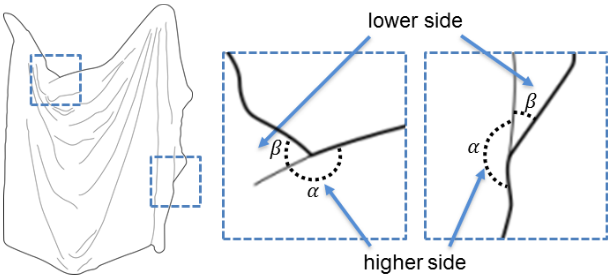
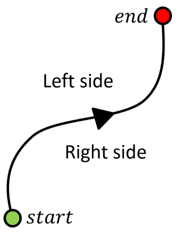


Fig. 6. Perceptual Cue: T-junction.

On a wrinkled surface, different surface regions could deform to different extents. Hence, a line drawing can contain ridge/valley strokes [17], [18], suggestive contours [19], as well as occluding strokes [20], [21] (see Fig. 5), depending on the amount of depth difference, or gradient across the strokes. Our simplified stroke model describes a smooth transition across these depth profiles using a signed value denoted as $g_i \in [-1, +1]$ for stroke s_i. The sign of g_i indicates the gradient direction, or equivalently which side of the stroke is higher (or lower) in the depth profile, while its magnitude indicates the relative amount of depth difference. Moreover, we define the orientation reference (left and right sides) of each stroke by traveling along the stroke from its start to end point (see the inset figure); here, g_i is *positive* if the *left side* of the stroke is *higher*, and vice versa. This model can cover various notable features in line drawings, since a large portion of human-drawn lines [23] are used for conveying local features like occlusion, ridges, and valleys.



Next, we explore five different perceptual cues to obtain local hints for estimating g_i (Sec. 4.2), and then formulate an optimization model to combine these local hints to infer a globally-consistent stroke interpretation (Sec. 4.3).

4.2 Perceptual Cues

We present five perceptual cues, each considering a different perceptual aspect to help estimate the local gradient (g_i): *T-junction* considers a local occlusion; *convexity* considers the shape of individual stroke; *continuity* considers a stroke pair that appears to continue from each other like a single curve; *proximity* considers a stroke pair that appears to press on each other to form a bulge or dent; and *regularity* considers a group of evenly-spaced strokes of similar shapes.

i) T-junction. A T-junction is formed when an endpoint of a stroke lies on another stroke. Perceptually, this suggests a local occlusion, see the *amodal completion law* [52]: "we tend to interpret the interrupted curve as the boundary of some object undergoing an occlusion." E.g., in the blown-up views shown in Fig. 6, the side next to the interrupted curve is lower than the side next to the other curve.

We explored two metrics for T-junction cue proposed by Liu et al. [55] and Yeh et al. [56]. We adopt the latter one, since we found it to be more robust for our case in experiments. In detail, we compute the T-junction cue as:

$$T_i = \frac{1}{\pi/2}\Big[\ |(\alpha \bmod \pi) - \frac{\pi}{2}| - |(\beta \bmod \pi) - \frac{\pi}{2}|\ \Big]\ , \quad (1)$$

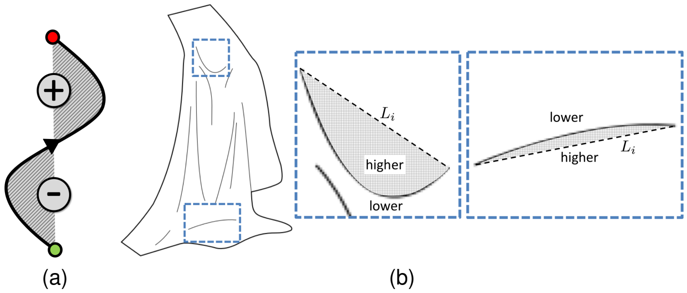


Fig. 7. (a) Inflections give rise to ambiguity in perceiving the depth across a stroke. (b) Perceptual Cue: Convexity.

where α and β are the junction angles (see again Fig. 6) and mod is the modulus operator. The range of T_i is $[-1,+1]$, where a value close to zero indicates a weak T-junction (e.g., when $\alpha \approx \beta$), while a value close to +1 (or -1) indicates a strong T-junction with the side on angle α (or β) being higher than the other. Note that T_i is set to zero for strokes that are not associated with any T-junction.

ii) Convexity. In Gestalt psychology [62], convex shapes could be perceptually associated to figure or foreground regions. Several works have used this cue to separate figure and ground [63], and to estimate local layering [12], [54]. In wrinkles, see Fig. 7b, when a stroke bends towards one side, this suggests a local bulge in the associated convex region. Hence, the bulged side of the stroke would be higher than the other. However, if the stroke contains an inflection (see Fig. 7a), ambiguity could arise in the perception.

From these observations, we quantify the convexity cue based on the deviation (signed area) of a stroke (s_i) from the straight line (L_i) that joins s_i's endpoints (see again Fig. 7a):

$$V_i = \texttt{clamp}(\ A_i/(\eta A_0)\ ,\ [-1,+1]\ )\ , \tag{2}$$

where A_i is the signed area of the stroke deviated from L_i (see shaded areas in Fig. 7a&b), A_0 is the area of the semicircle with L_i as its diameter, η is a parameter set to be 0.5 in all our experiments, and $\texttt{clamp}(\texttt{value}, [\texttt{min}, \texttt{max}])$ truncates the given value into range $[\texttt{min}, \texttt{max}]$. Using this formulation, strokes with inflections receive small $|V_i|$, while convex strokes with single-sided bending receive large $|V_i|$. Again, a positive V_i indicates that the left side of the stroke is higher, and vice versa.

iii) Continuity. As suggested by the Gestalt principle of good continuation [52], [64], human tends to perceive a curve as continuing along its established direction. Hence, strokes that appear to extend from each other are often perceptually seen as a single stroke, see Fig. 8; in other words, they would appear to have similar depth profiles or gradients.

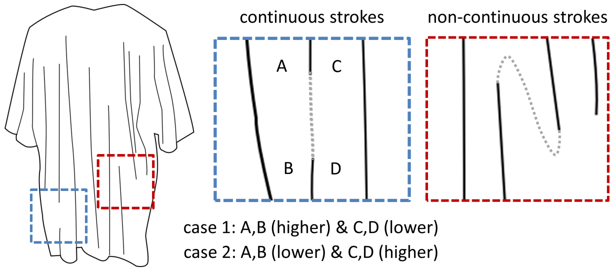

continuous strokes
non-continuous strokes
A
C
B
D
case 1: A,B (higher) & C,D (lower)
case 2: A,B (lower) & C,D (higher)

Fig. 8. Perceptual Cue: Continuity.

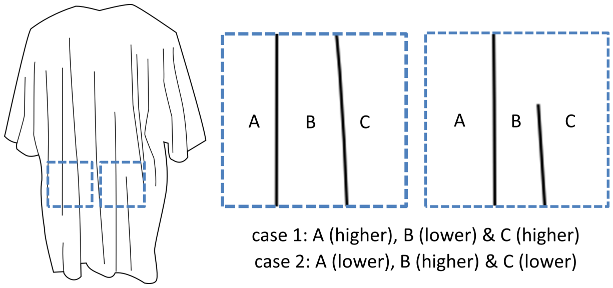

A
B
C
A
B
C
case 1: A (higher), B (lower) & C (higher)
case 2: A (lower), B (higher) & C (lower)

Fig. 9. Perceptual Cue: Proximity.

To estimate the continuity between a pair of strokes, say s_i and s_j, we first find a pair of stroke endpoints, one from each stroke, such that the distance between the two endpoints is the shortest among the four possible endpoint pairs. We then extract a small segment of stroke near each endpoint, and fit a cubic spline that passes through the two endpoints as well as overlaps the two segments, see the dashed lines in Fig. 8 (right). Next, we sum up the total absolute curvature $K_{i,j} = \int |k(s)| ds$ along the spline, where k is the curvature and s is the arclength parameter, and quantify the continuity term:

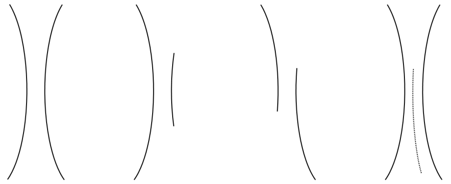
$$C_{i,j} = G(0; \frac{\pi}{4})(K_{i,j})\ , \tag{3}$$

where $G(\mu;\sigma)(\cdot)$ is a Gaussian kernel with mean μ and standard deviation σ for controlling the fall-off. Hence, G helps to normalize $C_{i,j}$ to $[0,1]$, such that when $K_{i,j}$ is zero, $C_{i,j}$ is one, and when $K_{i,j}$ increases, $C_{i,j}$ will gradually drop towards zero accordingly. Also note that we set $C_{i,j}$ to zero if the length of the spline is longer than the summed length of the two strokes (s_i and s_j), or the spline crosses (intersects) some other strokes in the line drawing.

iv) Proximity. For wrinkles illustrated in line drawings, it is common to see nearby strokes (nearly parallel) that appear to press on each other. Hence, the region in-between would perceptually look like a bulge or dent resulted from a local deformation, see the T-shirt example shown in Fig. 9. Note that the bulges and dents occur alternatively across the regions around the associated strokes, see cases 1 and 2 illustrated in Fig. 9. See also the inset figure, which is the reference photo that the artist [65] employed for sketching Fig. 9; label D indicates a dent, while label B indicates a bulge.

From observation, the strength of proximity cue (or the local deformation) is affected by: i) the spatial proximity between strokes, i.e., the closer the stronger; ii) the parallelity of stroke pair; iii) the perception is reduced (or even nullified) if some other stroke intercepts in-between; and iv) the perception is unaffected even though the length of the two strokes are unequal (see the right box in Fig. 9). To model the proximity cue with these characteristics, commonly-used metrics such as Hausdorff distance [66] and Fréchet distance [67] would perform poorly, especially for (ii)-(iv), see Fig. 10 for a quantitative comparison.

Our metric is computed as follows. Given strokes s_i and s_j, without loss of generality, we assume that s_i is not longer than s_j. Then, we sample N uniformly-spaced points along



H:	0.533	0.457	0.583	0.533
F:	0.653	0.616	0.601	0.653
1-P:	0.428	0.308	0.425	0.810

Fig. 10. Comparing metrics for quantifying proximity in various forms of wrinkle stroke pairs commonly found in line drawings: Hausdorff distance (H), Fréchet distance (F), and ours (1-P), where P is our proximity metric. Note that proximity is inversely proportional to the distance. The left three cases should be a strong proximity (low values) while the rightmost case should be weak (high values) due to occlusion.

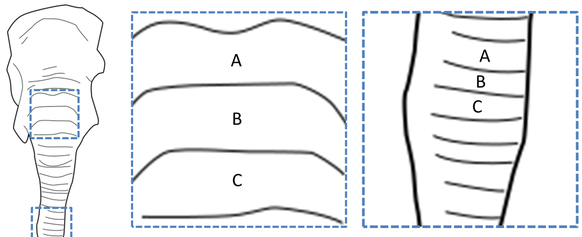
s_i, say p_k with $k \in [1, N]$, where p_1 and p_N are endpoints of s_i. For each p_k, we determine a point q_k on s_j, such that the distance between p_k and q_k, i.e., $||p_k - q_k||$, is minimized. Next, we define a binary function δ_k, which is zero if the straight line segment that joins p_k and q_k intersects some other strokes, and one otherwise. From these, we compute the proximity cue $P_{i,j}$ (s_i w.r.t. s_j) by averaging the distance values modulated by δ_k and a Gaussian kernel:

$$P_{i,j} = \frac{1}{N} \sum_{p_k \in s_i} \delta_k \cdot G(0; 0.2) \left(\min_{q_k \in s_j} \|p_k - q_k\| \right) , \tag{4}$$

where G is a Gaussian kernel with μ=0 and σ=0.2 for mapping distance values to $[0, 1]$. Note that if the point pair, p_k and q_k, are near and unoccluded, the mapped value will be close to one, and vice versa. For normalization purpose, we scale the input drawing, so that its larger side has one-unit length while its aspect ratio is preserved.

v) Regularity. When three or more similarly-shaped strokes align roughly in the same orientation with similar spacing in-between, see Fig. 11, we tend to perceive them as a group with similar property [52]. In this situation, the strokes in the group would appear to portray similar depth profiles rather than alternating bulges and dents.

We consider the following criteria in modeling the regularity cue: i) similarity of gap size in between adjacent strokes, ii) the local depth profile of the strokes as suggested by the convexity cue (V_i) and T-junction cue (T_i), and iii) rough alignment of the strokes along the same orientation. Before we formulate the cue, we first construct a stroke graph $\mathbf{G}$ to model the spatial proximity among the strokes in the line drawing. Each stroke (s_i) is represented as a node in $\mathbf{G}$. An edge is added between nodes, say s_i and s_j, if $P_{i,j}$ is larger than a threshold (which is set to be 0.5) and the two strokes have similar depth profiles as suggested by local cues V_i and T_i. For each edge, we compute $l_{i,j}$, the spacing (average distance) between strokes s_i and s_j.



case 1: A higher than B and B higher than C
case 2: A lower than B and B lower than C

Fig. 11. Perceptual Cue: Regularity.

Next, we aim to find subgraphs with two or more edges in $\mathbf{G}$ that have similar spacing sizes ($l_{i,j}$). This can be done by a simple breadth-first traversal from each edge in $\mathbf{G}$. For each subgraph obtained, we compute the regularity cue for each of its connecting edges:

$$R_{i,j} = G(0; 0.2)(|\ l_{i,j} - \bar{l}\ |) , \tag{5}$$

where G is a Gaussian kernel to put $R_{i,j}$ into range $[0, 1]$, and $\bar{l}$ is the median of $l_{i,j}$ among the edges in the subgraph. In this way, $R_{i,j}$ measures how well each pair of strokes fits other strokes within the same regularity group. For stroke pairs not in any regularity group, we set $R_{i,j}$ = 0.

4.3 Optimization Model

Energy Terms. Now, we formulate an energy term for each perceptual cue to constrain the unknown variables g_i (which is the local gradient of stroke s_i; see Sec. 4.1) accordingly:

- *T-junction Energy E_T* measures the deviation of g_i from the associated T-junction cue estimation (T_i):

$$E_T = \sum_i |T_i| \cdot \|g_i - T_i\|^2 . \tag{6}$$

 By multiplying with $|T_i|$, we can achieve the following: if a stroke associates with a strong T-junction, $|T_i|$ will be large, so it will be harder for g_i to deviate from T_i, since our optimization model (to be presented later) minimizes E_T. On the other hand, if a stroke associates with a weak/ambiguous T-junction or simply does not associate with any T-junction, $|T_i|$ is small or zero, so g_i can be more freely adjusted and may take a value further from T_i.

- *Convexity Energy E_V* is formulated like E_T for measuring the deviation of g_i from the associated V_i:

$$E_V = \sum_i |V_i| \cdot \|g_i - V_i\|^2 . \tag{7}$$

- *Continuity Energy E_C* encourages a stroke pair (say s_i and s_j) with high continuity value ($C_{i,j}$) to have similar depth profiles (local gradients g_i and g_j), since they would perceptually appear to be a single stroke:

$$E_C = \sum_{i,j\ s.t.\ i \neq j} C_{i,j} \cdot \phi(s_i, s_j) \cdot (g_i \cdot g_j) , \tag{8}$$

 where ϕ takes a value of -1 or +1 to rectify the sign of g_i and g_j (see the inset figure in Sec. 4.1); ϕ is +1 if the orientations of s_i and s_j do not match at the nearby endpoints, and -1 otherwise. Note also that we multiply ϕ, g_i and g_j, so that when we minimize E_C in the optimization, g_i and g_j are encouraged to have the same sign (both positive or both negative) if s_i and s_j have the same orientation, or opposite signs if s_i and s_j have different orientations.

- *Proximity Energy E_P* encourages a stroke pair s_i and s_j with high proximity $P_{i,j}$ to have different signs, so that they can have different depth profiles, resulting as alternating bulges and dents:

$$E_P = \sum_{i,j\ s.t.\ i \neq j} P_{i,j} \cdot (-\phi(s_i, s_j)) \cdot (g_i \cdot g_j) . \tag{9}$$

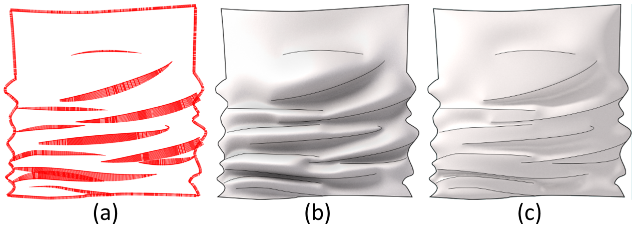


Fig. 12. Inflation with wrinkles. We set up a sparse gradient field based on the estimated local gradients across the strokes (left), and employ it to further estimate the 3D surface by computing a dense gradient field. The surface will be reconstructed to be smooth for soft shading (middle), and flat on the higher side of strokes for flat shading (right).

Note that E_P is formulated in a way similar to E_C, but we use $-\phi$, so that we can encourage local gradients g_i and g_j (of strokes s_i and s_j, resp.) to have opposite signs and produce alternating depth profiles when $|P_{i,j}|$ is large.

- *Regularity Energy* E_R encourages strokes in a regularity group to have similar depth profiles (or local gradients g_i) based on their affinity ($R_{i,j}$) to the group. We formulate this as a pairwise energy term similar to E_C:

$$E_R = \sum_{i,j \ s.t. \ i \neq j} R_{i,j} \cdot \phi(s_i, s_j) \cdot (g_i \cdot g_j) \ . \tag{10}$$

Putting the terms together. To obtain a globally-consistent interpretation of the strokes, we combine and balance the estimations from individual cues (see Sec. 6 for our comparison experiment) and seek a solution $\{ \ g_i \ : \ g_i \in [-1,1]; \ \forall i \ \}$ that minimizes the combined energy terms:

$$\min \left\{ \lambda_T E_T + \lambda_V E_V + \lambda_P E_P + \lambda_C E_C + \lambda_R E_R \right\}, \tag{11}$$

where λ_T, λ_V, λ_P, λ_C, and λ_R are weights for balancing the influences of the energy terms. Since T-junction and regularity cues are empirically found to be stronger than the other cues, we set λ_T=λ_R=8, whereas λ_V=2 and λ_P=λ_C=5. In addition, since the objective function is quadratic but non-convex, we solve it by sequential quadratic programming [68], which iteratively approximates and solves the objective function with a quadratic programming subproblem. In our experiments, the optimization usually converges in $\sim$ 5 to 20 iterations.

5 Inflation with Wrinkles

After estimating the depth profile of each stroke, we next need to recover partial 3D information to support the shading computation. To this end, we develop a wrinkle-aware inflation method to lift up the 2D drawing to 3D in three steps, as explained in Sections 5.1 to 5.3, followed by the shading process presented in Sec. 5.4.

5.1 Sparse Stroke Gradients

First, we define the surface gradient along each stroke using the estimated depth profiles g_i. Since the gradient resulted from this step will influence the shading style to be achieved from the partial geometry, we consider two cases:

Case (i) For 3D-like soft shading, the goal is to reconstruct a smooth and fair surface that will imbue a 3D-look to the line drawing when shaded. In this case, we define the gradient along each stroke as follows:

1) Compute 2D unit normal vectors $\mathbf{n}(s)$ sampled along each stroke s_i (to the left side of s_i) in 2D image space, where s is the arclength parameter.
2) Compute sparse 2D gradient vectors along the strokes as $F^*(x,y) = w(s) \cdot g_i \cdot \mathbf{n}(s)$, where $w(s)$ is a weight function along each stroke to smoothly attenuate the gradient vectors from the middle of the stroke to its endpoints, see the length of the little red gradient vectors along each stroke in Fig. 12a. In detail, we model $w(s)$ as a cubic polynomial with $w(0)$=$w(1)$=0 and $w(1/3)$=$w(2/3)$=1. To exclude a T-junction, we set $w(0)$ or $w(1)$ to one, if the associated endpoint is a junction.

Case (ii) For flat shading, the shading regions have hard boundaries and usually do not go across the strokes. While hard boundaries can be achieved by thresholding, it is challenging to ensure that the shading region does not go across the strokes. Our observation is that if the higher side of the stroke is inflated extensively, it may cast shadow/shading that goes across the stroke. Hence, we avoid inflating the higher side when defining the sparse stroke gradients as follows:

1) For strokes (s_i) whose local gradient $|g_i|$ is larger than a threshold (set as 0.25), we duplicate it into two, say s_i^{h} and s_i^{l}, in the image space of the drawing, and associate s_i^{h} and s_i^{l} respectively to the higher and lower sides of the local region around the stroke based on the sign of g_i. Hence, we can constrain the gradient field independently on the left and right sides of the strokes.
2) Compute the gradient along each stroke using steps 1 & 2 in Case (i) above, and assign the result to s_i^{l}.
3) Define sparse 2D gradient vectors along s_i^{h} as $F^*(x,y) = [0,0]^T$, so that the higher side of the reconstructed height field would be flat near the stroke.

Boundary. For both cases (i) and (ii), we set the gradient vectors along boundary $\partial\Omega$, using inward-pointing normals: $F^*(x,y) = (-b(s) \cdot \mathbf{n}(s))$, where $b(s)$ modulates the normal vectors $\mathbf{n}(s)$, so that we can control the local shape near the boundary of the partial geometry (surface) to be reconstructed. For example, if $b \approx 0$, the surface is locally flat. In our experiments, we set b as a constant based on the perceived local shape near the contour. Additionally, we smoothly attenuate $b(s)$ near the lower side of the T-junctions using $w(s)$ as in Case (i) above.

5.2 Dense Gradient Field

Our second step is to obtain a dense gradient field $F(x,y)$ over the 2D object region Ω, using the sparse gradients defined in Sec. 5.1. We propagate these sparse gradient vectors by minimizing the following energy functional:

$$\min_F \int_\Omega |\nabla F(x,y)|^2 + |\nabla \times F(x,y)|^2 \, dA \tag{12}$$

subject to:

$$F(x,y) = F^*(x,y), \quad \forall (x,y) \in \{s_i\} \cup \partial\Omega \ ,$$

where ∇ is the gradient operator, $\nabla\times$ is the curl operator, and dA is an infinitesimal area in Ω. The first term seeks a dense gradient field $F(x,y)$ that smoothly interpolates the sparse gradient vectors defined along the strokes and the boundary. Note that this is equivalent to seeking a zero Laplacian, i.e., $\Delta F(x,y) = 0$. The second term minimizes the curl, so that F is integrable. Visually, minimizing the curl ensures that the constrained gradients from F^* are retained as such in F, and the gradient vectors spatially propagate farther throughout the object region (Ω).

5.3 Surface from Gradient Field

Next, we aim to recover an approximate height field f from the gradient field F. Ideally, $\nabla f = F$, but F may not be integrable. Hence, we apply the divergence operator and seek for $\Delta f = \nabla \cdot F$ by a minimization:

$$\min_f \int_\Omega |\Delta f - \nabla \cdot F|^2 + \lambda |f - f^*|^2 dA\,. \tag{13}$$

subject to:

$$f = f^*, \quad \forall f \in \{s_i\} \cup \partial\Omega\,,$$

where the first term seeks surface f whose gradient approximates F, and the second term constrains f to remain close to f^* (a height function) along strokes and boundary; this can be set to 0 for objects that are flat overall. Otherwise, f^* is estimated by solving the Poisson equation: $-\Delta f^*(x,y) = c$, $\forall (x,y) \in \Omega \setminus \partial\Omega$ subject to Dirichlet, i.e., fixing the height function to 0 at the boundary to simulate a bulge near the boundary, and/or Neumann boundary conditions, i.e., fixing the normal derivative of the height function to 0 to simulate thin deformable material such as cloth. In this equation, c controls f^* similar to the inflation model employed by Sýkora et al. [10] and λ controls how close to f^* we would like f to be; we set it as a small value in range $[0, 0.001]$. The user may opt for Neumann boundary condition along the deformable portions of the boundary, e.g., see the bottom boundary of the cloth in Fig. 1 (left). In this way, we can estimate a wrinkled surface f as a height field for computing the shading, see Fig. 12(b).

Discretization Details

We solve Eqs. 12 & 13 numerically by triangulating Ω into a mesh [69] with the following constraints: mesh edges pass through strokes in the drawing, and mesh faces have similar areas and angles. We discretize the differential operators at each mesh vertex v_i, by standard linear finite element formulations based on one-ring neighbors $\mathcal{N}(v_i)$ to reduce Eqs. 12 & 13 to systems of linear equations. Specifically, the Laplacian at each vertex v_i is defined as [70], [71]:

$$\Delta(v_i) \approx \frac{1}{A_i} \cdot \frac{1}{2} \sum_{v_j \in \mathcal{N}(v_i)} (\cot\alpha_{ij} + \cot\beta_{ij})(v_j - v_i)\,,$$

where α_{ij} and β_{ij} are angles opposite to the edge v_i-v_j on either side, and A_i is the Voronoi area at v_i that accounts for the differences in mesh sampling.

The discrete curl used in Eq. 12 is defined as [72]:

$$\nabla \times F(v_i) \approx \sum_{v_j, v_k \in \mathcal{N}(v_i)} F_{ijk} \cdot e_{jk}\,,$$

where the sum is taken over triangles v_i-v_j-v_k incident to v_i, e_{jk} refers to the edge vector from v_j to v_k, and F_{ijk} is the per-face average value of the vector field F.

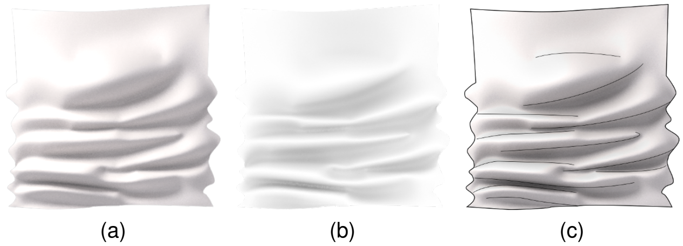


Fig. 13. 3D-like soft shading style. We first generate a shading image rendered with the Blinn-Phong shading model (a), followed by an ambient occlusion map (b). These are combined with the original line drawing being overlaid to generate the final soft shading result (c).

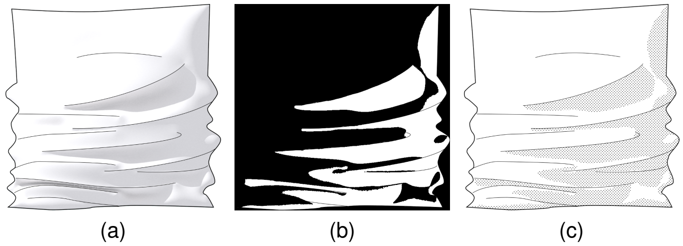


Fig. 14. Flat shading in manga style. We begin with an initial shading (a), and generate a binary mask (b), on which we apply a screentone texture to obtain a manga-style flat shading result (c).

The discrete divergence in Eq. 13 is defined as [73]:

$$\nabla \cdot F(v_i) \approx \frac{1}{2} \sum_{v_j, v_k \in \mathcal{N}(v_i)} \cot\theta_{ij}(e_{ij} \cdot F_{ijk}) + \cot\theta_{ki}(e_{ki} \cdot F_{ijk})\,,$$

where the sum is taken over triangles v_i-v_j-v_k incident to v_i, and θ_* refers to angles opposite to the associated edge e_*.

5.4 Shading

The height field f resulted in Sec. 5.3 provides partial 3D information for determining the shading regions. Our inflation method reconstructs 3D meshes that adhere to the characteristics of 3D-like soft shading and flat shading. This enables us to generate these shading styles using conventional rendering tools. To produce the rendering, we first set up a directional light, and use an orthographic camera. The resulting shading image can be multiplied with, or overlaid on the original line drawing to produce the shaded result, while keeping the image features originally in the input drawing.

For 3D-like soft shading, we may use an off-the-shelf global illumination engine to render the inflated mesh. We generate such a result (Fig. 13c) by computing the shading with the Blinn-Phong model and soft shadowing (Fig. 13a), and an ambient occlusion map (Fig. 13b) using 3dsMax 2016.

For manga-style flat shading, we first generate the shading image as above while setting the diffuse color to white, and computing hard shadows (Fig. 14a). Next, we convert it to a binary image by Otsu's thresholding [74] to obtain shading regions with hard boundaries (Fig. 14b). Lastly, we apply a screentone texture on the shading regions to generate a manga style shading effect (Fig. 14c).

6 Results & Discussion

6.1 Implementation

We implemented and ran our method on a desktop computer with a 3.2GHz Intel Core i7 processor and 12 GB

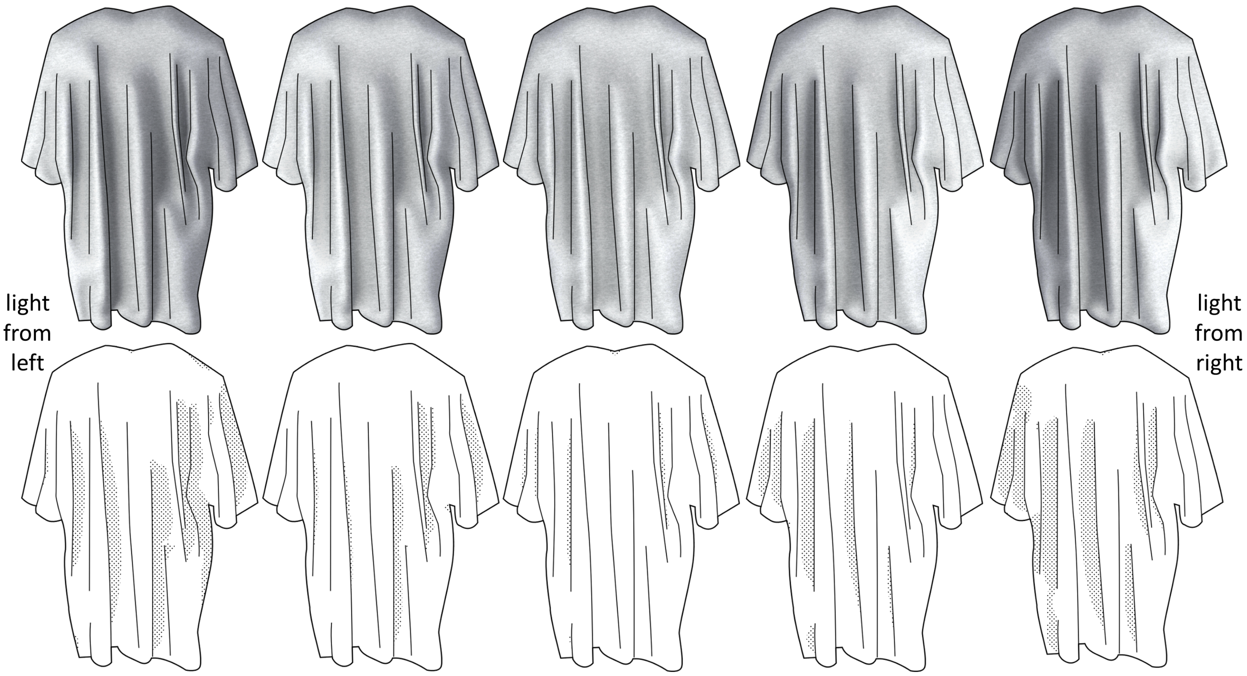


Fig. 15. 3D-like soft shading (top) and flat shading in manga style (bottom) rendered with a directional light source moving from left to right.

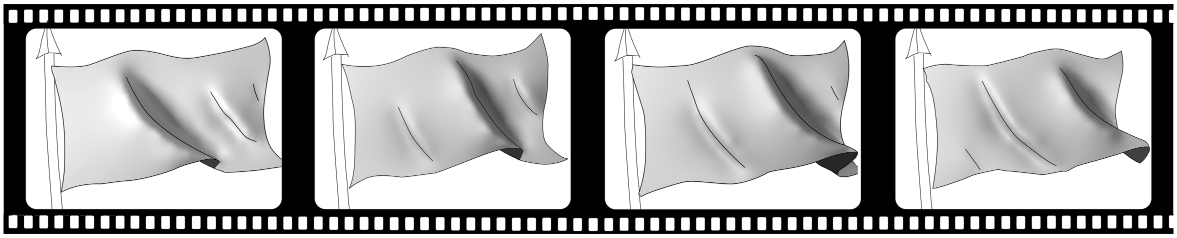


Fig. 16. Shading results on an animation sequence of a waving flag (obtained from [75]) in 3D-like soft shading style. The line drawing in each frame was extracted and vectorized as input to our method that constructs the partial geometry. Note that the light source is on the left side.

TABLE 1
Time taken to generate shading from the line drawings.

Input	Stroke Analysis	Optimization	Inflation	
			# Triangles	Time
Cloth	11.6s	0.012s	41,600	84.86s
T-shirt	35.8s	0.024s	32,543	64.10s
Tablecloth	39.5s	0.031s	30,326	67.33s
Jean	171s	0.106s	29,438	60.71s
Gown	264s	0.118s	33,806	72.83s
Dress	75.44s	0.095s	29,052	57.36s
Towel	150.9s	0.065s	41,693	85.72s

memory. Our method takes a clean line drawing as input, along with user-defined parameters, e.g., balancing weights in optimization, boundary conditions, c in the Poisson equation for f^*, the direction of the light source, and the shading style; their default values were mentioned earlier. Then, our method automatically performs stroke analysis, global optimization, inflation, and shading. Table 1 shows the timing statistics for various line drawings. Note that we did not intentionally optimize our implementation for best performance, and will leave it as our future work.

6.2 Shading Results

Fig. 17 presents our shading results on input drawings with various forms of wrinkle strokes. Column 2 shows global illumination renderings of the reconstructed partial geometry, columns 3 & 4 show results in 3D-like soft shading style, while columns 5 & 6 show results in manga-style flat shading. We put the light source on top left for results in columns 3 & 5, and on top right for columns 4 & 6. We may also change the light source direction and modify the shading without recomputing everything in the pipeline. Fig. 15 shows a shading result sequence generated by gradually moving the light source from left to right.

Next, we apply our method to shade image frames in a line drawing animation by extracting and vectorizing frames in an animation; see Fig. 16, which shows that the shading is fairly smooth across the frames. Note that we did not consider temporal coherency in this work, since it is a nontrivial problem on its own. To do this, we may interpolate certain intermediate results in our method, e.g., local gradients and partial geometry. Interpolating the partial geometry is challenging, since we may not be able to ensure smoothly varying gradients around the strokes over the animation frames. On the other hand, interpolating local stroke gradients over the frames seems to be simpler, but topological changes of the strokes in the line drawing can lead to a stroke correspondence problem, which could be hard to handle. We leave this problem as a future work.

Note that to produce the results shown in the paper, we use the default values of the parameters, except for a few: i) c for finding f^* in Eq. 13, since c describes the global shape of the given object, ii) λ in Eq. 13, and iii) $b(s)$ for controlling the boundary, based on our perception of the 3D shape.

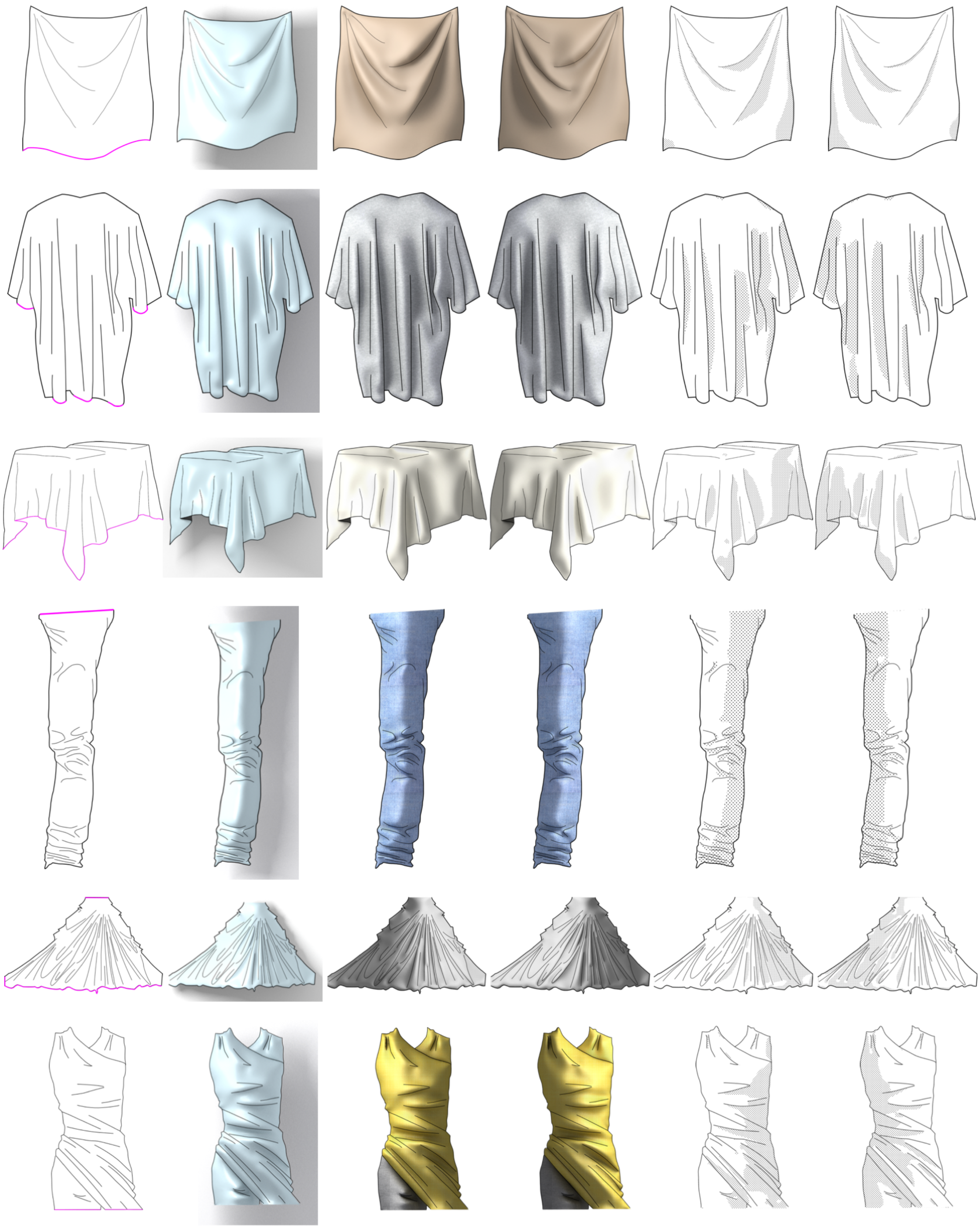


Fig. 17. Shading results generated by our method: CLOTH, SHIRT, TABLECLOTH, JEAN, GOWN, and DRESS (from top to bottom). Column 1: input line drawings; Neumann boundary conditions are revealed in magenta. Column 2: surfaces (the partial geometry) rendered with glossy reflections and global illumination. Columns 3 & 4: 3D-like soft shading results. Columns 5 & 6: manga-style flat shading results. Note that the light source is fixed at top-left for Columns 3 & 5 and at top-right for Columns 4 & 6, respectively.

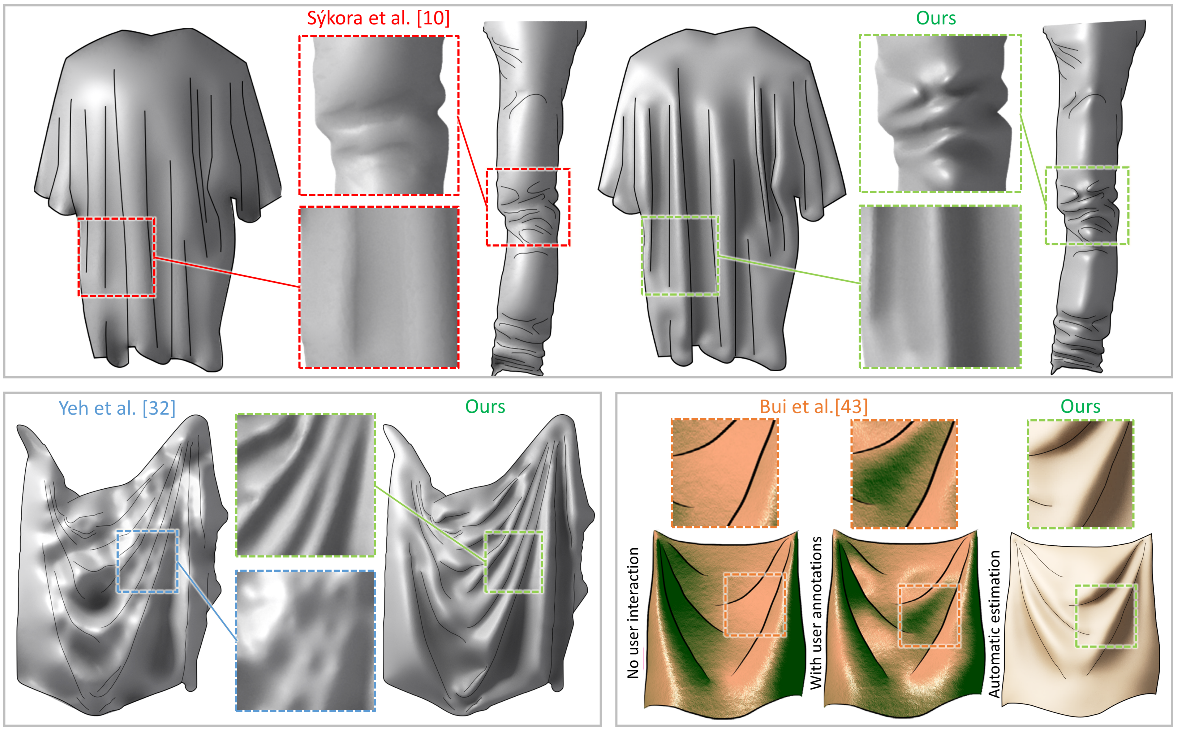


Fig. 18. Comparison with three closely-related recent works. In detail, we render the geometry kindly generated and provided by Dr. Sýkora [10] (top gray box) and Dr. Yeh [32] (bottom-left gray box), and take the results from Bui et al. [43] (bottom-right gray box) for comparison.

6.3 Comparison with Closely-related Methods

We compare our method with three closely-related recent works on reconstructing 3D information from a single input image; see Fig. 18. In particular, we compare the visual details in the shading results for illustrating wrinkles.

Sýkora et al. [10] built an interactive framework to reconstruct bas-relief meshes from images of line drawings and to support the generation of global illumination effect. Their boundary inflation method alone cannot handle wrinkles, and can only generate convex or concave shapes based on the region boundaries. However, their framework allows users to segment the wrinkle regions, and inflate each segment into convex (bulge) or concave (dent) shapes, and integrate the result into a single surface. We passed the input drawings to the authors and obtained from them the reconstructed surfaces; see Fig. 18 (top). Note that the number of segments and inflation parameters are crucial for good results with wrinkles. Hence, this approach generally requires tedious works for complex drawings, such as JEAN and TOWEL with crowded wrinkle strokes. On the other hand, our method explicitly models the interior strokes, and hence, it does not require additional manual works for region segmentation and surface integration. We also found that visually, our method can generate more detailed geometry; see the blown-up views in Fig. 18 (top).

Yeh et al. [32] proposed an interactive approach to generate high-relief geometry from a single input image by accepting user annotations at specific points on the image to constrain the local slope and curvature. Similarly, we obtained results generated from their method, and present one of them in Fig. 18 (bottom left). This approach proved to be tedious for generating wrinkled height fields, since their point-based annotations (rather than stroke-based) are not effective in generating smooth wrinkled surfaces. While we may reconstruct the global shape of an object by sparse annotations, adding local details around wrinkle strokes require dense and tedious user annotations in their framework. In contrast, our results are smooth and plausible for generating shading, without user annotation.

Bui et al. [43] designed a user-driven method to generate shading on line drawings using a hatching metaphor: users could sketch hatching patterns on the drawing, and provide cues about the local normal orientation. In the absence of such information, their method smoothly interpolates the normals from the contour lines without considering the local spatial structure and interaction among neighboring strokes. Therefore, the wrinkle regions are ill-defined in the geometry; see Fig. 18 (bottom right, left subfigure). However, by additionally incorporating user-drawn hatching strokes, their method is able to generate a plausible normal field; see Fig. 18 (bottom right, center subfigure). In contrast, by analyzing the spatial arrangements of strokes in the line drawing, our method can automatically estimate a consistent geometry from the line drawing without user annotations; see Fig. 18 (bottom right, right subfigure). Moreover, since our inflation method reconstructs a height field surface, rather than just a normal field, we can generate various visual effects, e.g., ambient occlusion, shadows, global illumination, etc.

6.4 Evaluation: Global Optimization

We evaluate how the global optimization model contributes by comparing partial geometries estimated from (i) local cues alone and (ii) combined cues after global optimization. For local estimation, we take the maximum value of the T-junction and convexity cues as the depth profile of each stroke, since the other cues are constraints over multiple strokes. For global estimation, we apply our method in

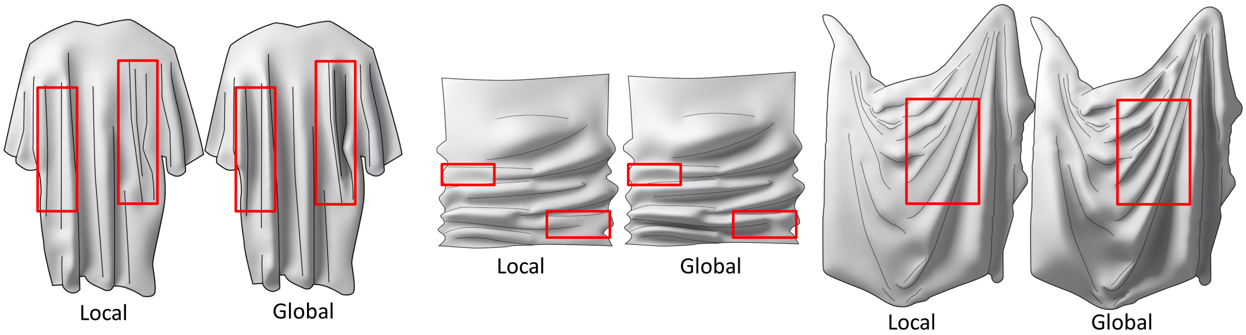


Fig. 19. Comparing partial geometry results generated from local estimations (maximum of T-junction and convexity cue) and global estimation (the global optimization method in Sec. 4.3), corresponding to the left and right images, respectively, in each subfigure. From the red boxes, we can see that the optimization can propagate the local gradient profiles, thus helping to generate richer and plausible shading results with more details.

Sec. 4.3 to combine and balance the local cues. We then apply the inflation method in Sec. 5 to generate the partial geometries from these local and global estimations.

From the results in Fig. 19, we can see that the local cues alone may not be sufficient to create perceptually plausible wrinkle geometry. Consider the leftmost red box in Fig. 19; these strokes depict a wrinkle, but since they are nearly straight lines, we cannot determine the depth profile across these strokes from the local cues. On the other hand, the depth profile of the same strokes in the third red box from the left are derived from global optimization, where the continuity and proximity cues help infer and propagate the depth profile of the strokes, resulting in plausible wrinkle geometry. Similarly, we can find richer wrinkle geometry in the second and third comparison cases shown in Fig. 19 (middle & right).

7 Conclusion

This paper presents a novel method to analyze wrinkle strokes in clean line drawings for automated shading production. The key contributions include: i) computational models of five perceptual cues (T-junction, convexity, proximity, continuity, and regularity) for wrinkle strokes, in which we explored the local shape, spatial relationship, and interaction of strokes by consulting relevant psychological principles; ii) an optimization formulation for deriving a globally consistent interpretation of wrinkle strokes by combining and balancing local estimations from the cues; and iii) a wrinkle-aware inflation method for generating a partial geometry to support two commonly-used shading styles. Compared to the prior work, our method has several unique features such as the ability to automatically generate rich wrinkle geometry conveyed through the interior strokes, infer a globally consistent stroke interpretation, and support shading characteristics employed by artists with the various examples shown in the results. These features facilitate the generation of plausible shading even on line drawings with crowded wrinkle strokes, which cannot be handled by the previous works.

Limitations. Our method shares some common limitations with other works on single-image 3D reconstruction techniques. First, even though our method can generate plausible shadings in a given view, it may not reconstruct a perfect 3D surface when the actual (perceived) depth depicted by the strokes in the drawing vary drastically (e.g., see TABLECLOTH). Second, we assume that the geometry is smooth and non-polyhedral, i.e., without sharp edges and tilted planes. Third, our current implementation uses a fixed-height or free boundary condition along the boundary contours with constant gradients, since this work focuses mainly on the analysis of interior wrinkle strokes. Fourth, we assume clean line drawings as inputs, where the strokes are drawn purposely for depicting certain features without errors, e.g., strokes that intersect and cross one another. Lastly, we only consider lighting and geometry as the main influencing factors for shading, and did not take into account other artistic criteria; see future work below.

Future work. First, it is worth pointing out that there are other factors that could affect the shading style, e.g., the semantics of the scene and the artists' personal preference or style. In future, we plan to extend our method with more shading styles and characteristics, e.g., to learn the artist drawing styles, and to explore automatic and smart methods for line drawings with regions of different semantics. Second, we are interested in extending our optimization and inflation framework to support shading of animated line drawings, typically by considering temporal consistency [55] for ensuring smoothly varying shading results. Third, we currently assume the input lines to be of one-pixel width. This assumption could be relaxed in future by considering lines of varying width, as well as their hierarchical organization in the drawing, since artists associate such fine lines with features that are less prominent. Lastly, we are interested in exploring the inclusion of user interactivity for handling very complex line drawings, e.g., drawings that depict both smooth and polyhedral shapes, drawings with folded [40] or double-sided surfaces [32], and drawings with nonlinear boundary conditions that require additional user intervention in reconstructing the partial shape.

Acknowledgments

We thank the anonymous reviewers for the various suggestions and comments. We are grateful to Dr. Daniel Sýkora and Dr. Chih-Kuo Yeh for their generous help in generating results for the comparison. This research is partially supported by the BeingTogether Centre, a collaboration between Nanyang Technological University, Singapore and University of North Carolina, Chapel Hill, and The Chinese University of Hong Kong strategic recruitment fund and direct grant (4055061).

References

[1] A. Macworth, "Interpreting pictures of polyhedral scenes," *Artificial Intelligence*, vol. 4, no. 2, pp. 121–137, 1973.

[2] I. Chakravarty, "A generalized line and junction labeling scheme with application to scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, no. 2, pp. 202–205, 1979.

[3] K. Sugihara, "Mathematical structures of line drawings of polyhedrons-Toward man-machine communication by means of line drawings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 4, no. 5, pp. 458–469, 1982.

[4] J. Malik, "Interpreting line drawings of curved objects," *Intl. J. Comput. Vis.*, vol. 1, no. 1, pp. 73–103, 1987.

[5] H. Lipson and M. Shpitalni, "Optimization-based reconstruction of a 3D object from a single freehand line drawing," *Computer-Aided Design*, vol. 28, no. 8, pp. 651–663, 1996.

[6] P. A. C. Varley and R. R. Martin, "Estimating depth from line drawing," in *Proc. ACM Symp. Solid Modeling and Applications*. ACM, 2002, pp. 180–191.

[7] T. Igarashi, S. Matsuoka, and H. Tanaka, "Teddy: A sketching interface for 3D freeform design," in *Proc. SIGGRAPH*, 1999, pp. 409–416.

[8] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa, "FiberMesh: Designing freeform surfaces with 3D curves," *ACM Trans. Graph. (SIGGRAPH)*, vol. 26, no. 3, pp. 41:1–41:10, 2007.

[9] Y. Gingold, T. Igarashi, and D. Zorin, "Structured annotations for 2D-to-3D modeling," *ACM Trans. Graph. (SIGGRAPH Asia)*, vol. 28, no. 5, pp. 148:1–148:9, 2009.

[10] D. Sýkora, L. Kavan, M. Čadík, O. Jamriška, A. Jacobson, B. Whited, M. Simmons, and O. Sorkine-Hornung, "Ink-and-Ray: Bas-Relief meshes for adding global illumination effects to hand-drawn characters," *ACM Trans. Graph.*, vol. 33, no. 2, pp. 16:1–16:15, 2014.

[11] wikiHow, wikihow.com/Draw-Fabrics, 2016, Last accessed 29 November 2016.

[12] S. F. Johnston, "Lumo: Illumination for cel animation," in *Intl. Symp. on Non-Photorealistic Anim. and Rendering*, 2002, pp. 45–ff.

[13] C. Shao, A. Bousseau, A. Sheffer, and K. Singh, "CrossShade: Shading concept sketches using cross-section curves," *ACM Trans. Graph. (SIGGRAPH)*, vol. 31, no. 4, pp. 45:1–45:11, 2012.

[14] E. Iarussi, D. Bommes, and A. Bousseau, "BendFields: Regularized curvature fields from rough concept sketches," *ACM Trans. Graph.*, vol. 34, no. 3, pp. 24:1–24:16, 2015.

[15] P. Joshi and N. A. Carr, "Repoussé: Automatic inflation of 2D artwork," in *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, 2008, pp. 49–55.

[16] J. Andrews, P. Joshi, and N. Carr, "A linear variational system for modelling from curves," *Comput. Graph. Forum (SGP)*, vol. 30, no. 6, pp. 1850–1861, 2011.

[17] V. Interrante, H. Fuchs, and S. Pizer, "Enhancing transparent skin surfaces with ridge and valley lines," in *Proc. IEEE Conf. Visualization*, 1995, pp. 52–59.

[18] Y. Ohtake, A. Belyaev, and H.-P. Seidel, "Ridge-valley lines on meshes via implicit surface fitting," *ACM Trans. Graph. (SIGGRAPH)*, vol. 23, no. 3, pp. 609–612, 2004.

[19] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella, "Suggestive contours for conveying shape," *ACM Trans. Graph. (SIGGRAPH)*, vol. 22, no. 3, pp. 848–855, 2003.

[20] L. Markosian, M. A. Kowalski, D. Goldstein, S. J. Trychin, J. F. Hughes, and L. D. Bourdev, "Real-time nonphotorealistic rendering," in *Proc. SIGGRAPH*, 1997, pp. 415–420.

[21] A. Hertzmann and D. Zorin, "Illustrating smooth surfaces," in *Proc. SIGGRAPH*, 2000, pp. 517–526.

[22] T. Isenberg, B. Freudenberg, N. Halper, S. Schlechtweg, and T. Strothotte, "A developer's guide to silhouette algorithms for polygonal models," *IEEE Comput. Graph. App.*, vol. 23, no. 4, pp. 28–37, 2003.

[23] F. Cole, A. Golovinskiy, A. Limpaecher, H. S. Barros, A. Finkelstein, T. Funkhouser, and S. Rusinkiewicz, "Where do people draw lines?" *ACM Trans. Graph. (SIGGRAPH)*, vol. 27, no. 3, pp. 88:1–88:11, 2008.

[24] L. Olsen, F. Samavati, and J. Jorge, "NaturaSketch: Modeling from images and natural sketches," *IEEE Comput. Graph. App.*, vol. 31, no. 6, pp. 24–34, 2011.

[25] B. Xu, W. Chang, A. Sheffer, A. Bousseau, J. McCrae, and K. Singh, "True2Form: 3D curve networks from 2D sketches via selective regularization," *ACM Trans. Graph. (SIGGRAPH)*, vol. 33, no. 4, pp. 131:1–131:13, 2014.

[26] C. De Paoli and K. Singh, "SecondSkin: Sketch-based construction of layered 3D models," *ACM Trans. Graph. (SIGGRAPH)*, vol. 34, no. 4, pp. 126:1–126:10, 2015.

[27] L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge, "Sketch-based modeling: A survey," *Computers & Graphics*, vol. 33, no. 1, pp. 85–103, 2009.

[28] M. T. Cook and A. Agah, "A survey of sketch-based 3-D modeling techniques," *Interacting with Computers*, vol. 21, no. 3, pp. 201–211, 2009.

[29] J. Assa and L. Wolf, "Diorama construction from a single image," *Comput. Graph. Forum (Eurograph.)*, vol. 26, no. 3, pp. 599–608, 2007.

[30] S. Iizuka, Y. Endo, Y. Kanamori, J. Mitani, and Y. Fukui, "Efficient depth propagation for constructing a layered depth image from a single image," *Comput. Graph. Forum (Pacific Graph.)*, vol. 33, no. 7, pp. 279–288, 2014.

[31] Q. Zeng, W. Chen, H. Wang, C. Tu, D. Cohen-Or, D. Lischinski, and B. Chen, "Hallucinating stereoscopy from a single image," *Comput. Graph. Forum (Eurograph.)*, vol. 34, no. 2, pp. 1–12, 2015.

[32] C.-K. Yeh, S.-Y. Huang, P. K. Jayaraman, C.-W. Fu, and T.-Y. Lee, "Interactive high-relief reconstruction for organic and double-sided objects from a photo," *IEEE Trans. Vis. Comput. Graph.*, vol. PP, no. 99, 2016, to appear.

[33] Y. Wang, Y. Chen, J. Liu, and X. Tang, "3D reconstruction of curved objects from single 2D line drawings," in *Proc. IEEE Conf. Comput. Vis. Pattern Rec.*, 2009, pp. 1834–1841.

[34] T. Xue, J. Liu, and X. Tang, "Example-based 3D object reconstruction from line drawings," in *Proc. IEEE Conf. Comput. Vis. Pattern Rec.*, 2012, pp. 302–309.

[35] D. Sýkora, D. Sedlacek, S. Jinchao, J. Dingliana, and S. Collins, "Adding depth to cartoons using sparse depth (in)equalities," *Comput. Graph. Forum (Eurograph.)*, vol. 29, no. 2, pp. 615–623, 2010.

[36] J. Hahn, J. Qiu, E. Sugisaki, L. Jia, X.-C. Tai, and H. S. Seah, "Stroke-based surface reconstruction," *Numerical Mathematics: Theory, Methods and Applications*, vol. 6, no. 01, pp. 297–324, 2013.

[37] O. A. Karpenko and J. F. Hughes, "SmoothSketch: 3D free-form shapes from complex sketches," *ACM Trans. Graph. (SIGGRAPH)*, vol. 25, no. 3, pp. 589–598, 2006.

[38] M. Kolomenkin, G. Leifman, I. Shimshoni, and A. Tal, "Reconstruction of relief objects from archeological line drawings," *J. Comput. Cult. Herit.*, vol. 6, no. 1, pp. 3:1–3:19, 2013.

[39] K. Rose, A. Sheffer, J. Wither, M.-P. Cani, and B. Thibert, "Developable surfaces from arbitrary sketched boundaries," in *Comput. Graph. Forum (SGP)*, 2007, pp. 163–172.

[40] A. Jung, S. Hahmann, D. Rohmer, A. Begault, L. Boissieux, and M.-P. Cani, "Sketching folds: Developable surfaces from non-planar silhouettes," *ACM Trans. Graph.*, vol. 34, no. 5, pp. 155:1–155:12, 2015.

[41] T.-P. Wu, C.-K. Tang, M. S. Brown, and H.-Y. Shum, "ShapePalettes: Interactive normal transfer via sketching," *ACM Trans. Graph. (SIGGRAPH)*, vol. 26, no. 3, pp. 44:1–44:5, 2007.

[42] Q. Xu, Y. Gingold, and K. Singh, "Inverse Toon Shading: Interactive normal field modeling with isophotes," in *Proc. Workshop on Sketch-Based Interfaces and Modeling*, 2015, pp. 15–25.

[43] M. T. Bui, J. Kim, and Y. Lee, "3D-look shading from contours and hatching strokes," *Computers & Graphics (SMI)*, vol. 51, no. C, pp. 167–176, 2015.

[44] Y. Qu, W.-M. Pang, T.-T. Wong, and P.-A. Heng, "Richness-preserving manga screening," *ACM Trans. Graph. (SIGGRAPH Asia)*, vol. 27, no. 5, pp. 155:1–155:8, 2008.

[45] K. Lawonn, A. Baer, P. Saalfeld, and B. Preim, "Comparative evaluation of feature line techniques for shape depiction," in *Vision, Modeling & Visualization*, 2014, pp. 31–38.

[46] K. Lawonn and B. Preim, *Feature lines for illustrating medical surface models: Mathematical background and survey*. Springer International Publishing, 2016, pp. 93–131.

[47] F. Cole, K. Sanik, D. DeCarlo, A. Finkelstein, T. Funkhouser, S. Rusinkiewicz, and M. Singh, "How well do line drawings depict shape?" *ACM Trans. Graph. (SIGGRAPH)*, vol. 28, no. 3, pp. 28:1–28:9, 2009.

[48] K. A. Stevens, "The visual interpretation of surface contours," *Artificial Intelligence*, vol. 17, no. 1, pp. 47–73, 1981.

[49] J. J. Koenderink, "What does the occluding contour tell us about solid shape?" *Perception*, vol. 13, no. 3, pp. 321–330, 1984.

[50] D. Knill, "Perception of surface contours and surface shape: From computation to psychophysics," *J. Optical Soc. of Amer.*, vol. 9, no. 9, pp. 1449–1464, 1992.

[51] R. Arnheim, *Art and visual perception: A psychology of the creative eye*. University of California Press, 1974.

[52] A. Desolneux, L. Moisan, and J.-M. Morel, *From Gestalt theory to image analysis: A probabilistic approach*, 1st ed. Springer, 2007.

[53] Y. Ming, H. Li, and X. He, "Connected Contours: A new contour completion model that respects the closure effect," in *Proc. IEEE Conf. Comput. Vis. Pattern Rec.*, 2012, pp. 829–836.

[54] G. Palou and P. Salembier, "Monocular depth ordering using T-junctions and convexity occlusion cues," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1926–1939, 2013.

[55] X. Liu, X. Mao, X. Yang, L. Zhang, and T.-T. Wong, "Stereoscopizing cel animations," *ACM Trans. Graph. (SIGGRAPH Asia)*, vol. 32, no. 6, pp. 223:1–223:10, 2013.

[56] C.-K. Yeh, P. Jayaraman, X. Liu, C.-W. Fu, and T.-Y. Lee, "2.5D cartoon hair modeling and manipulation," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 3, pp. 304–314, 2015.

[57] L. Nan, A. Sharf, K. Xie, T.-T. Wong, O. Deussen, D. Cohen-Or, and B. Chen, "Conjoining Gestalt rules for abstraction of architectural drawings," *ACM Trans. Graph. (SIGGRAPH Asia)*, vol. 30, no. 6, pp. 185:1–185:10, 2011.

[58] X. Liu, T.-T. Wong, and P.-A. Heng, "Closure-aware sketch simplification," *ACM Trans. Graph. (SIGGRAPH Asia)*, vol. 34, no. 6, pp. 168:1–168:10, 2015.

[59] L. Zhang, H. Huang, and H. Fu, "EXCOL: An EXtract-and-COmplete layering approach to cartoon animation reusing," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 7, pp. 1156–1169, 2012.

[60] getty, http://getty.deviantart.com/art/Where-are-we-25302128, Last accessed 29 November 2016.

[61] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, pp. 1–28, 2016.

[62] G. Kanizsa, W. Gerbino, and M. Henle, "Convexity and symmetry in figure-ground organization," *Vision and Artifact, Springer*, pp. 25–32, 1976.

[63] H.-K. Pao, D. Geiger, and N. Rubin, "Measuring convexity for figure/ground separation," in *Intl. J. Comput. Vis.*, vol. 2, 1999, pp. 948–955.

[64] S. Ullman, "Filling-in the gaps: The shape of subjective contours and a model for their generation," *Biological Cybernetics*, vol. 25, no. 1, pp. 1–6, 1976.

[65] M. Bérubé, how-to-draw-funny-cartoons.com/how-to-draw-clothes.html, Last accessed 29 November 2016.

[66] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, 1993.

[67] T. Eiter and H. Mannila, "Computing discrete Fréchet distance," Technische Universität Wien, Tech. Rep., 1994.

[68] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4, pp. 1–51, 1995.

[69] J. R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and delaunay triangulator," in *Applied Comp. Geom.*, ser. Lecture Notes in Comp. Sci. Springer-Verlag, 1996, vol. 1148, pp. 203–222.

[70] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. SIGGRAPH*, 1999, pp. 317–324.

[71] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, *Visualization and Mathematics III*, 2003, ch. Discrete differential-geometry operators for triangulated 2-manifolds, pp. 35–57.

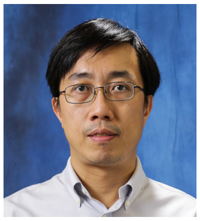
[72] F. do Goes, M. Desbrun, and Y. Tong, "Vector field processing on triangle meshes," in *SIGGRAPH Asia 2015 Courses*, 2015, pp. 17:1–17:48.

[73] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in Heat: A new approach to computing distance based on heat flow," *ACM Trans. Graph. (SIGGRAPH)*, vol. 32, no. 5, pp. 152:1–152:11, 2013.

[74] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[75] C. Hages, "2D Animation roughs_ Flag Flap," youtube.com/watch?v=qtEgUxjF6GQ, Last accessed 15-Feb-2017.

Pradeep Kumar Jayaraman is currently a Ph.D. student in the School of Computer Science and Engineering at Nanyang Technological University, Singapore. He received his B.Tech degree in Information Technology from Anna University, India in 2009. His research interests lie at the intersection of computer vision, computer graphics and numerical optimization.



Chi-Wing Fu recently joined the Chinese University of Hong Kong as an associate professor in 2016. He obtained his PhD in Computer Science from Indiana University Bloomington USA. He is currently serving as the program co-chair of SIGGRAPH ASIA 2016 technical brief and poster, and he has served as members in various program committees, including SIGGRAPH ASIA technical brief, emerging technology, IEEE visualization, and ACM CHI Work-in-Progress. He served as associate editor of Computer Graphics Forum, and received the IEEE Transactions on Multimedia Prize Paper Award (an annual award from the IEEE Signal Processing Society). His research interests include computer graphics, visualization, and user interaction research in HCI.

Jianmin Zheng is an associate professor in the School of Computer Science and Engineering at Nanyang Technological University, Singapore. He received the BS and PhD degrees from Zhejiang University, China. His recent research focuses on T-spline technologies, digital geometric processing, virtual reality, visualization, interactive digital media and applications. He has published more than 150 technical papers in international conferences and journals. He has served on the program committee of many international conferences. Currently he is an associate editor of The Visual Computer.

Xueting Liu received her B.Eng. degree from Tsinghua University and Ph.D. degree from The Chinese University of Hong Kong in 2009 and 2014 respectively. She is currently a postdoctoral fellow in the Department of Computer Science and Engineering, The Chinese University of Hong Kong. Her research interests include computer graphics, computer vision, computational manga and anime, and non-photorealistic rendering.

Tien-Tsin Wong received his B.Sc., M.Phil. and Ph.D. degrees in Computer Science from The Chinese University of Hong Kong in 1992, 1994, and 1998 respectively. He is currently a professor in the Department of Computer Science and Engineering, The Chinese University of Hong Kong. His main research interests include computer graphics, computational manga, precomputed lighting, image-based rendering, GPU techniques, medical visualization, multimedia compression, and computer vision. He received the IEEE Transactions on Multimedia Prize Paper Award 2005 and the Young Researcher Award 2004.