Animated Construction of Line Drawings

Hongbo Fu — City University of Hong Kong
Shizhe Zhou* — Zhejiang University
Ligang Liu — Zhejiang University
Niloy J. Mitra — KAUST/UCL

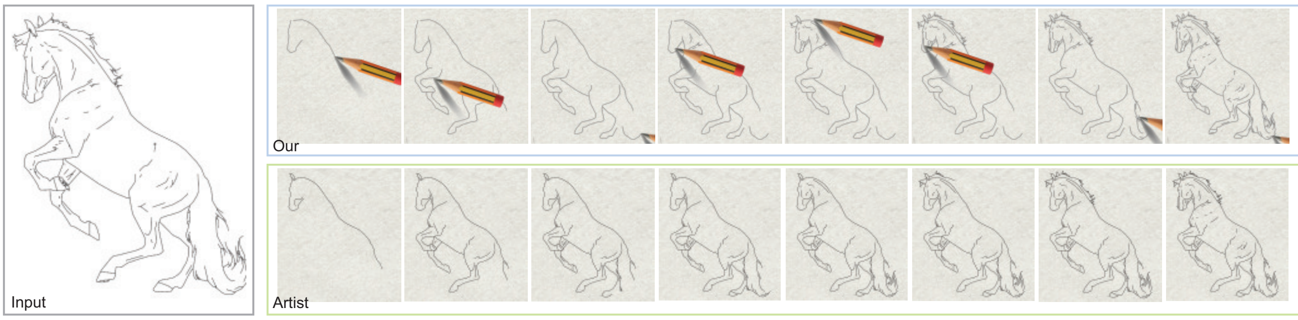


Figure 1: *Guided by sketching principles, we derive a plausible stroke order of an input static line drawing (left) to automatically animate the sketching (right top). A user study shows that the inferred order is comparable to the order used by an artist (right bottom).*

Abstract

Revealing the sketching sequence of a line drawing can be visually intriguing and used for video-based storytelling. Typically this is enabled based on tedious recording of artists' drawing process. We demonstrate that it is often possible to estimate a reasonable drawing order from a static line drawing with clearly defined shape geometry, which looks plausible to a human viewer. We map the key principles of drawing order from drawing cognition to computational procedures in our framework. Our system produces plausible animated constructions of input line drawings, with no or little user intervention. We test our algorithm on a range of input sketches, with varying degree of complexity and structure, and evaluate the results via a user study. We also present applications to gesture drawing synthesis and drawing animation creation especially in the context of video scribing.

Keywords: line art, animation, drawing analysis, video scribing

Links: DL PDF WEB

1 Introduction

Line art is a popular art form, and is widely used for illustrations, caricatures, cartoons, etc. Demonstration videos and commercials (e.g., a series of animations by the RSA) often use animation sequences showing the drawing process of line artworks as a mode of instruction to vividly tell stories. The technique for producing such dynamic line art (with synchronized audio content) is often referred as *video scribing*, which is desirable for building anticipation, directing viewer attention from one object to another, conveying order of action sequences in instructional animations, or simply for maintaining continuity during narrative storytelling. Traditionally, video scribing animations are created by recording the drawing process during the creation of line art images, either by using video cameras or recording functionalities of drawing software. This limits the creation of video scribing to a group of professional drawers.

Our work is motivated by millions of searchable line drawings on the Internet and intends to enable ready reuse of available drawings for video scribing. Given a static line art image, we intend to estimate a drawing order from the image itself, which is visually plausible to a human viewer (see Figure 1). Basic drawing principles [Guptill and Meyer 1997; Willats 1997] naturally demand a solution to the following key problems (see Figure 2): (i) construct a coarse-to-fine hierarchical representation of an input line drawing image, since drawers, both amateurs and professionals, mostly start with a rough sketch and then gradually refine the drawing by introducing additional details, (ii) order a set of drawing strokes, and finally (iii) determine directions for each of the individual strokes.

The order of compilation of a drawing might vary with people or for the same person across time. This implies that our problems typically have multiple plausible solutions. Therefore, we focus on finding one of the *reasonable* solutions that look plausible to a human viewer, instead of searching for the *best* drawing order if any. Although drawing order of line art is subjective and involves personal taste and preferences, drawers indeed follow certain sets of rules due to their stereotypical behavior, as supported by extensive research studies in the cognitive psychology (see [van Sommers 1984; Tversky and Suwa 2009] and references therein).

We propose an effective solution for estimating a reasonable order given a number of 2D lines vectorized from line art images, with most of the curve lines being sharp and cleanly defined. Our work makes the following contributions: (i) introduce the problem of animated construction of line drawings, (ii) summarize geometric guidelines of drawing order from findings in drawing cognition, which are then computationally encoded through analysis of line drawings, and (iii) effectively simulate the drawing process by constructing a coarse-to-fine hierarchy and formulating the ordering of strokes as finding a Hamiltonian path on a graph encoding both the individual properties of lines (e.g., complexity) and their interrelations (e.g., proximity, collinearity, and anchoring).

Since the plausibility of the estimated drawing orders is subjective, we conduct a user study to evaluate the perceived quality of our so-

*Corresponding author: shizhezhou@zju.edu.cn

lutions compared with a set of alternative strategies. We analyze and evaluate a variety of line art images, including those created by artists and synthesized by computer graphics. We show the applications of animating line drawings to gesture drawing synthesis, and video scribing animation creation (see supplemental video).

2 Related Work

Art and cognition. Drawing-instruction books (e.g., [Dodson 1990; Nicolaides 1990; Edwards 1999]) seldom discuss specific principles of drawing order, but instead focus on visual representation of different drawing techniques. However, in psychology and cognitive science, the order of drawing elements of a sketch or design is important since the order reveals the underlying organization of the sketch [Tversky and Suwa 2009], across various levels (e.g., of a motor program, mental construction, or conceptual organization). In an important series of works, Novick and Tversky [1987; 1999] propose that the order in which line stokes are drawn reflects how we schematize and conceptualize objects. Further, they suggest that the drawing order of strokes also reveals our organization of a scene and its hierarchical representation. Note that although automatic drawing and painting systems without human intervention like AARON (cf. [Cohen 1995] and references therein) exist, such systems focus on the final drawing rather than the temporal aspect of the drawing process.

Writing order recovery. Observers can reliably extract drawing orders from static traces of handwritten forms [Babcock and Freyd 1988]. Solutions have been proposed to recover writing order from handwriting images, and use the knowledge to boost recognition rate (see [Nguyen and Blumenstein 2010] and references therein). Unlike drawing, writing usually has more specific ordering rules, which are almost uniquely defined for individual characters. Even then the problem of restoring *the correct writing* order, instead of *a reasonable drawing* order of a character, is rather challenging. Few attempts have been made in the context of multiple strokes [Kato and Yasuhara 1999] or logographic writing like Chinese characters [Wu et al. 2007]. Given the usual dependency across different parts of a sketch, our problem is significantly more complex.

Non-photorealistic rendering (NPR). Synthesizing line drawings from images and 3D models has been extensively studied [Winkenbach and Salesin 1994; Gooch and Gooch 2001; Strothotte and Schlechtweg 2002; Judd et al. 2007; Lee et al. 2007; Grabli et al. 2010]. Such techniques can be used to automatically produce compelling images as input to our framework. Mimicking hand drawing, House and Singh [2007] synthesize line drawings from 3D models by a dynamic process, with strokes generated by a physically-based moving pen. Their algorithm simply begins with a randomly selected point on the model and traverses feature edges based on connectivity. The drawing order produced by our algorithm can directly benefit such a system.

Northam et al. [2010] present a stroke-based renderer to reduce detail obstruction and enhance artistic styles for painterly rendering, e.g., for creating an image with a hand-painted appearance from a photography. They adopt the layer-based painterly rendering algorithm by Hertzmann [1998], which implicitly orders brush strokes by grouping strokes of similar salience into layers. Brush strokes at individual layers are then ordered by evaluating the values of *separate* stroke properties, e.g., in ascending/descending luminance order. In general their problem is more ill-posed than ours mainly due to large overlap between brush strokes.

Line drawing simplification. Our solution of hierarchy reconstruction is based on line drawing simplification. Early simplification algorithms focus on the application of density reduction, which is usually achieved by line omission [Grabli et al. 2004; Wilson and Ma 2004]: the importance of input lines is first evaluated mainly based on certain density measures and less important lines are then omitted to reduce clutter. Alternatively, Barla et al. [2005] present a simplification technique based on perceptual grouping, supporting multiple applications in a unified framework (e.g., density reduction, level-of-detail (LOD) representation, progressive drawing). Their technique iteratively clusters pairs of input lines, with grouping errors based on proximity and parallelism, and then replaces each group of lines with an application-oriented new line. Shesh and Chen [2008] propose a faster variant to support dynamic simplification.

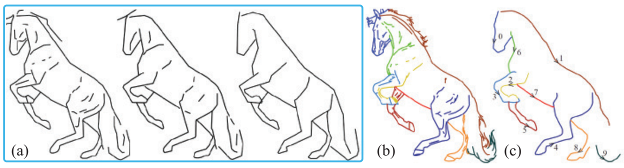


Figure 2: *Main stages of our framework: hierarchy reconstruction (a), grouping (b), and ordering on significant lines (c).*

Sketch understanding and recognition. Several research fields, such as sketch understanding [Sezgin et al. 2001] and engineering drawing analysis [Tombre 1998], often deal with the analysis of line drawings but a detailed review of them is beyond the scope of our paper. The solutions of those fields regularly use domain knowledge, e.g., mathematical symbols, mechanical engineering design, circuit diagrams, etc. for beautification and recognition of freehand sketches or engineering drawings.

Our work is orthogonal to efforts to map properties between a scene and a picture that represents the scene (i.e., the interaction between the two spaces), and optimize the final picture to best satisfy certain goals [Durand 2002; DeCarlo and Stone 2010] or studying how artists use line drawing for conveying 3D shapes [Cole et al. 2008]. In geometry processing, shape analysis has also been used for improved acquisition [Li et al. 2010] and for model abstraction [Mehra et al. 2009]. Instead we focus on deriving *temporal* information of drawing stokes, a problem which has largely been unexplored.

3 Overview

Input. Line art images consist of straight or curved lines to express the shape of 2D or 3D scene objects. In other words, line art emphasizes form and outline, over color, shading, or texture. Such rich geometric information is vital to our technique. In our system we consider only line art images with cleanly defined lines or curves, and exclude line drawings with shading or texture simulated by hatching or stippling, where individual lines are hard to distinguish from one another. We consider an input line drawing to be an image composed of an *unordered* set of *vectorized* 2D line strokes.

Broadly, line art is created in two styles: one with erasing and one without erasing. In the first scenario, a drawer begins with a rough sketch depicting a global structure of the drawing and then progressively introduces details by simply sketching over the existing strokes (also referred as over-sketching). The final drawing is obtained by erasing all intermediate reference lines. In the second scenario, all strokes drawn by the drawer directly contribute to and appear in the final artwork. The strokes are again inserted typically in a coarse-to-fine manner.

Hierarchy reconstruction. The above observations motivate us to first construct a coarse-to-fine hierarchy (Section 5.1), which serves as a *level-by-level* drawing order (Figure 2) for both these drawing styles, paves the way for the introduction of hierarchical drawing guidelines from drawing cognition (Section 4.1), and provides meaningful grouping of input line strokes (Figure 2b). Animating a line drawing in the second style (i.e., without erasing) is our main focus, for which we interpret the input line strokes as a two-level representation: the significant line strokes in each group form a coarse representation of the drawing and are drawn first (Figure 2c), followed by the rest of the lines representing the details.

Stroke ordering. In the second step, we estimate a drawing order of lines at individual levels (Section 5.2). We incorporate the guiding principles (Section 4) to order the significant lines and formulate the ordering problem as finding a Hamiltonian path in a directed graph (Section 5.3), with graph nodes corresponding to lines in the input drawing and graph edges carrying proximity and anchoring guidelines. Both graph nodes and edges are associated with appropriate cost based on the drawing guidelines such that the Hamiltonian path with the (approximate) minimum cost naturally corresponds to a reasonable drawing order. The order of the detail lines is achieved using a simplified ordering scheme. Finally, in Section 5.4, we determine the directions of the individual strokes.

4 Guidelines for Drawing Order

Based on extensive user studies, van Sommers [1984] prescribes various high-level guidelines for the order of drawing geometric designs and simple objects, which form the base of our research. In this section, we identify the important geometric guidelines.

The guidelines were originally obtained by carefully conducting and analyzing experimental studies of graphic productions, and provide insights to various layers of organization in the drawing performance of ordinary people (i.e., without professional drawing training), including adults and children. Although the guidelines were obtained via study and analysis of simple drawings, they are also applicable, as verified by our extensive evaluation, for ordering elements of complex drawings by carefully modeling the interaction among the individual simple guidelines (see Section 5).

In this work, we do not consider secondary effects arising from orientation of canvas surface, size of drawing, drawing medium, etc. For example, pens and brushes that need continual replenishment with ink impose an upper limit on the length of continuous strokes, significantly influencing the structure of the drawing process. There is no such issue with pencils or ballpoint pens. Since we start from a static trace, we simply assume that the input drawing is made with a pencil on A4-sized paper fixed to a horizontal surface, which is also one of the primary configurations for user studies adopted by van Sommers [1984].

4.1 Hierarchical structure of drawing

Drawings, being reflections of our mental abstraction [Novick and Tversky 1999], have a natural multi-layered structure. In general, (novice) subjects tend to draw elements from a coarse-to-fine manner. There are two alternative guidelines relating ordering of strokes with hierarchically organized forms:

H1 (Level by level): Subjects prefer to completely deal with one level in the hierarchy, before proceeding to the next level. For example, while drawing a human figure, arms are drawn first, then hands, then fingers.

H2 (Sub-hierarchy by sub-hierarchy): Subjects complete sub-hierarchies one by one. For example, they proceed to draw arm, hand, fingers on one side, and then do the same on the other side.

Given a hierarchical organization of line strokes, **H1** amounts to a breadth-first traversal of the hierarchy and the ordering of multiple units at each level is mainly dominated by geometric similarity. In contrast, **H2** is similar to a depth-first traversal, which can be considered as combined ordering based on proximity and semantics at the local level (within each sub-hierarchy) and similarity operations at the more global level (across sub-hierarchies).

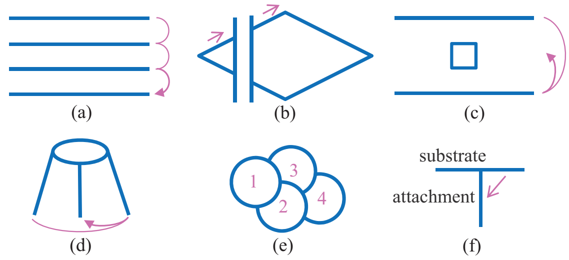


Figure 3: *Guidelines of order in stroke making: (a) by proximity, (b) by collinearity, (c) by similarity of form, (d) by symmetry, and (e,f) by anchoring. The order is shown with arrows or numbers.*

4.2 Order in stroke making

In each level of hierarchy, order among strokes is affected by a very wide spectrum of factors including geometric, semantic, and motor reflexes [Gombrich 1960; Viviani and Stucchi 1992]. Although semantic and motor convenience can impose important ordering constraints, especially in abstract sketches, we mainly focus on the geometric aspects given a static trace as input. However, by favoring continuity, e.g., treatment of T-junctions, we do favor longer strokes even across gaps as favored by motor convenience.

According to geometric considerations, other things being equal, both right- and left-handed subjects group components on the basis of five factors, as follows (see Figure 3):

R1 (Simplicity): Most start by drawing lines with simple shapes or smooth lines. Subsequently, these lines are used as guides for more complex curves.

R2 (Proximity): Proximity plays a dominant role in determining order and starting positions of intermediate stokes. Specifically, subjects invariably move to the nearest element or the element along the trajectory they are following, due to obvious economies of effort and decision.

R3 (Collinearity): For geometric and semantic reasons, drawers favor continuity, and thus follow the trajectory of interrupted lines.

R4 (Similarity): Subjects intend to group drawing elements in the executive process by their similarity of size, form, orientation, etc. The progression from one to another for elements of a similar type moves in a consistent order, e.g., clockwise from 11 o'clock to 5 o'clock, top to bottom, or left to right.

R5 (Symmetry): Symmetrical elements are repeated without any other drawing acts intervening. The order of drawing these repeated units is similar to that based on similarity.

Another geometric constraint affecting the order of compilation of a drawing is *anchoring* or *end control* of lines, which involves the attachment of a new stroke to a substrate, i.e., existing structures. For example, the crossbar (substrate) of a T-junction is almost invariably laid down first before the stem (attachment), as shown in Figure 3f. The basis of anchoring preference lies in the ease of control to achieve accuracy since it is easier to locate a (stationary) pencil at rest at the beginning of a stroke. Without anchoring, the drawers have to anticipate the location of future end points and intercept end points of completed arcs with moving lines, which is difficult to control. Generally, subjects favor working from front to back or from intact to occluded (Figure 3e and supplementary video). The corresponding anchoring guideline suggests:

R6 (Anchoring): Substrate strokes are almost invariably drawn first for the attachment of other connected strokes.

4.3 Stroke direction

Having determined a line order, the next step in drawing is to decide in which direction to move the pencil. Such preferences naturally depend on the convenience of mechanical movement of the drawer's wrist and fingers, and visibility of drawn strokes, making right- and left-handers behave differently. The guidelines for stroke direction are presented next.

D1 (Preferred directions): Downward vertical strokes and left-to-right horizontal strokes are favored by right-handed subjects, with left-handers showing an analogous mirror-image preference.

D2 (Rarely to upper-left): Both right- and left-handed subjects generally avoid drawing lines towards top-left directions.

When producing simple geometric strokes, e.g., sets of straight lines, the majority of strokes conform to the preferred stroke directions. However, for complex strokes with high-curvature corners, it is common that drawers turn a corner without lifting the pencil off the paper (i.e., continuous paper contact is favored) and may then have to proceed in an otherwise non-preferred direction. The primary motive behind this is economy since it reduces the number of executive commands while maintaining accuracy at line intersections without requiring separate location controls. Thus,

D3 (Paper contact): Non-preferred directions may be taken to favor continuous paper contact.

4.4 Starting location

The remaining decision is where on the paper to begin, i.e., the problem of *starting location*. User studies indicate that the preference of starting location is irrespective of the drawers' handedness:

S1 (Starting location): Most drawers favor a starting position near the top part of a stroke of a simple drawing, e.g., containing simple geometric forms. Interestingly, the preferences of starting location and stroke direction are largely independent.

5 Methodology

In this section, we map the guidelines summarized in the previous section into computational procedures.

Preprocessing. Our system accepts as input a set of unordered line strokes, vectorized from line drawing images (obtained using publicly available *WinTopo* software for our examples). Each stroke curve is encoded as a one-parameter curve, sampled uniformly along its arc length. The software outputs only open curves, breaking loops as necessary. For inputs with smudge lines or strong hatching effects, we manually identify and delete the corresponding curves, typically taking less than a minute of user interaction.

5.1 Hierarchy reconstruction

Almost any drawing involves hierarchical organization among its constituent elements. We first analyze the input set of curves and organize them into a hierarchy with the input curves being the fine scale features, which are grouped together by coarse scale proxies capturing the global structure of the line drawing (Figures 2 and 4). From the coarse to fine level, the hierarchy itself serves as a level-by-level drawing order of the input drawing. A natural drawing order involves starting with a rough skeletal structure, and then progressively introducing additional lines for filling in geometric details. Motivated by this, we introduce an iterative simplification approach that alternates between two steps: (i) a *clustering* step for reducing the number of curves, and (ii) an *approximation* step for reducing shape complexity of individual strokes.

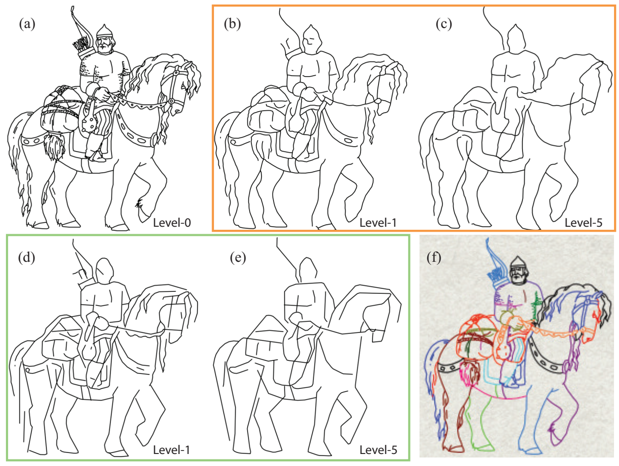


Figure 4: *(a) Input drawing; (b,c) extracted hierarchy using the modified algorithm of Barla et al. [2005]; (d,e) extracted hierarchy by our alternating simplification approach; (f) final grouping result.*

The clustering step is mainly based on the line drawing simplification algorithm introduced by Barla et al. [2005], which iteratively clusters pairs of lines with a minimum error measure. Their definition of error metric takes *proximity* and *continuity*, effectively enforcing desired guidelines **R2** and **R3**. In order to avoid complex curves while combining pairs of clustered lines to new ones, the algorithm introduces ε-lines and ε-groups to enforce parallelism among the new lines with the initial ones, where ε is a parameter for controlling the simplification scale (see [Barla et al. 2005] for details). We make two modifications to the original algorithm. First, we use a length-weighted interpolation scheme for interpolating a new line from a pair of lines, thus encouraging the long curves to remain close to their original positions. Second, we relax the definitions of ε-lines and ε-groups and allow small portions of lines, 5% in our examples, to violate the original definitions, which eases the choice of the parameter ε.

Instead of simply preserving the shape of the original drawing during clustering as in the algorithm of Barla et al. [2005], we focus on progressively reducing shape complexity of lines (**R1**). To achieve this, we create polyline approximations of the curves. For a given curve, we find its piecewise linear approximation and require the polyline vertices to lie on the curve, which is a standard problem of curve polygonization [Rosin 2002] with parameter ζ controlling the approximation error. Progressively reducing the number and simplifying the shape of lines are equivalent to choosing increasing values of simplification scale parameter ε and approximation error parameter ζ. Thus, for generating a hierarchy among the input lines, we apply a series of alternating simplifications with progressively increasing ε and ζ, each time starting from the previous, finer level. Figure 4 shows typical simplification results. Such constructed hierarchy can be used to produce over-sketching animation (see supplemental video), where the coarse level of the hierarchy serves as guiding lines for the whole drawing.

Two-level representation. Note that, by construction each line in the coarse level of the hierarchy corresponds to a group of lines in the original drawing, thus providing a meaningful grouping across the input lines (Figures 2b and 4f). Each group often has a multi-layered structure: one or more lines, which we call the *significant* lines (e.g., Figure 2c), depict the global shape; while the rest of the lines serving as the details, which we call the *detail* lines, anchor to the significant lines. In each group, we mark lines as significant based on their length (other priority ordering can also be used). First, we leave out the groups whose longest lines have length smaller than 10% of the diagonal of the drawing's bounding box. For each of the remaining groups, we first pick the longest line l as one of its significant lines. We sort the remaining lines based on their length. We mark them as significant if they are not close to any of the already selected significant lines (according to Hausdorff distance) and their length is larger than $\lambda\|l\|$, where $\|l\|$ denotes the length of the line l and $\lambda \in [0, 1]$ ($\lambda = 0.7$ in our implementation). Let $L = \{l_i | i = 0, 1, \ldots, n-1\}$ denote the set of significant lines from *all* the groups, encoding the global structure of the input drawing (see Figures 2c and 8).

5.2 Stroke ordering formulation

Guided by sketching principles (Section 4), we now order the existing strokes of the input. Following **R6** and **H1**, we first draw all the significant lines, followed by the detail ones. We give preference to **H1** over **H2**, because **H2** demands the evaluation of high-level semantic similarity between sub-hierarchies, which is rather challenging to automatically extract, especially from small approximate hand-drawn sketch lines. Next, we describe how to assign a drawing order, i.e., temporal order, to the significant lines in set L. The ordering of the detail lines will be discussed at the end of Section 5.3.

A simple solution is to greedily impose a drawing ordering, as follows: given a starting line, the order of the rest of the lines can be progressively determined based on the current configuration and tracing history. The advantage of this scheme comes from its simplicity and low computational cost. It is, however, difficult to adapt such a heuristic scheme to various drawing styles (see Section 6.1). Instead we propose a *global* scheme by abstracting the ordering problem as a global energy minimization over a permutation ordering of index set $I = [0, 1, \ldots, n-1]$ with the corresponding energy minimized over all the permutations P of I. Note that at the level of structural lines, the number of lines is small enough to allow us to explore large parts of the combinatorial solution space, which we found to be crucial for deriving interesting orders and variations.

Energy function. We formulate the ordering problem as the following global minimization problem:

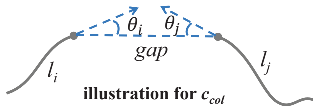
$$p^* := \operatorname*{argmin}_{p \in P} w \sum_{i=0}^{n-1} c_{ind}(l_{p_i})\theta(i) + \sum_{i=0}^{n-2} c_{tra}(l_{p_i}, l_{p_{i+1}}), \quad (1)$$

subject to additional constraints derived from detected anchoring configurations (see later). Here $c_{ind}(l)$ captures the properties of an individual line l (e.g., length and complexity etc.), $c_{tra}(l_i, l_j)$ evaluates the transition cost from line l_i to l_j (e.g., based on proximity and continuity), and w is a weight to balance the influence between the two terms (empirically, we found $w \in [1, 3]$ and set to $w = 1$ by default). The function $\theta(i)$ is a monotonically decreasing function ($\theta(i) = 1 - i/n$ in our implementation) to encourage the sorting of lines by their individual cost (e.g., to enforce the simplicity guideline (**R1**)). The sequence of $(l_{p_0^*}, l_{p_1^*}, \ldots, l_{p_{n-1}^*})$ gives us a desired ordering of L. Recall that our goal is to find a *reasonable* drawing order instead of recovering the original order that led to the input drawing.

Individual line cost. The individual line cost $c_{ind}(l)$ is simply measured as the *shape complexity* of a line according to **R1** in our implementation. Empirically, we arrived at the following measure to capture the desirable properties:

$$c_{ind}(l) = \eta\,(1 - \|\mathbf{v}_s - \mathbf{v}_e\|/\|l\|) + \sqrt{\textstyle\sum_{i=0}^{m-1} (\kappa_i - \bar{\kappa})^2/m}, \quad (2)$$

where $\mathbf{v}_s$ and $\mathbf{v}_e$ are the two end vertices of line l, m is the number of sampled vertices along l, κ_i and $\bar{\kappa}$ are the curvatures at vertex $\mathbf{v}_i$ and the mean value of $\{\kappa_i\}$, respectively, and η is a weight to balance the two terms ($\eta = 0.1$ in our experiments). The first term penalizes the deviation from a straight line, reaching zero for a straight line; while the second term measures the standard deviation of $\{\kappa_i\}$, reaching zero for constant-curvature lines like circles.



Transition cost. During sketching, transition from one line to another is affected by the inter-relations between the lines, namely proximity, collinearity, symmetry, and similarity (**R2-R5**). We observed that **R4** and **R5** are better at grouping lines instead of ordering the significant lines that are already representatives of each group. Further, the guidelines often contradict each other, making it challenging to combine them consistently. In our design, we focus on proximity **R2** and collinearity **R3**, expressed as follows:

$$c_{tra}(l_i, l_j) = w_p\, c_{pro}(l_i, l_j) + (1 - w_p) c_{col}(l_i, l_j), \quad (3)$$

where weight w_p (in the range [0, 1] and set to 1/9 as default) balances the two effects. We measure proximity $c_{pro}(l_i, l_j)$ as the distance between the closest points on l_i and l_j, and collinearity as $c_{col}(l_i, l_j) = \frac{gap}{\|l_i\| + \|l_j\|}(\theta_i + \theta_j)^2$ as the (positive) angular difference between endpoint tangents of l_i and l_j (see inset). Note that $c_{col}(l_i, l_j)$ allows relatively wider gap for pairs of longer lines. Both measures are normalized to $[0, 1]$.

Constraints. As an important coupling effect, anchoring guideline **R6** implies an *asymmetric* binary relation between two lines. For example, drawing of substrates (e.g., the crossbar of a T-junction) before attachments (e.g., the stem of a T-junction) involves advanced planning for anchoring. It does not, however, mean that the attachments should be drawn immediately after the substrate, i.e., it is common for a desired drawing order to have other lines in between the substrate and the attachment lines. Since formulating such inequality-like relation into $c_{tra}(l_i, l_j)$ is complicated, instead, we explicitly enforce such constraints during the optimization process, as explained later.

We found it sufficient to consider T-junctions for anchoring. Since, by construction, the lines in L are at comparable scales, we use the following simple strategy to define a T-junction (Figure 5): two lines l_s and l_c form the stem and crossbar of a T-junction, respectively if the following are satisfied: (i) l_c intersects with a short segment of the tangent line at one of the end vertices of l_s (Figure 5a); (ii) the (smaller) angle between l_c and the segment is larger than a threshold, e.g., 20^o (to exclude near parallel lines) and, (iii) the confidence of T-junction, $\min(s_1, s_2)/\|l_c\|$, is larger than a threshold, e.g., 5% (to avoid detecting corners as T junctions, see Figure 5b).

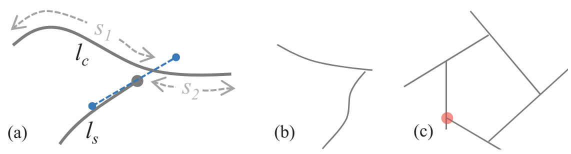


Figure 5: *(a) T-junction detection; (b) an example of corner instead of T-junction; (c) a T-junction cycle can be broken by rejecting a T-junction with least confidence as highlighted.*

5.3 Stroke ordering optimization

Solving the global minimization in Equation 1 is computationally expensive. A naïve solution involving simple enumeration of all the possible drawing orders is expensive $O(n!)$. Instead, we approximate the solution using a graph minimization, as described next.

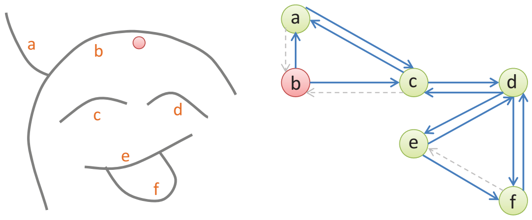


Figure 6: *(Left) Input sketch consisting of lines (a, b, ..., f). The red dot indicates the starting location of interest. (Right) Corresponding k-NN directed graph (here $k = 2$), with nodes representing input lines. Dashed arrows denote removed edges. Starting line l_b is highlighted. Since k-NNs are found for each node independently, nodes can have valence higher than k (see nodes c and d).*

Encoding R2. Since proximity (**R2**) has the primary influence on drawing order, we choose the next line to be drawn after the current line l as of one of its k-nearest neighbors (k-NN) of l. We encode the topological structure of the line drawing using a graph $G = (V, E)$, where each graph node in V represents a line in L, i.e., $V = L$ (Figure 6). The graph edges $E = \{(l_i, l_j)\}$ reflect the k-nearest neighborhood of the lines, i.e., if $(l_i, l_j) \in E$, l_j is a k-nearest neighbor of l_i, based on the distance metric used for $c_{pro}(l_i, l_j)$.

Encoding R6. In order to capture the anchoring constraints (**R6**), we make the graph G *directed*. To start with, all the current edges (based on k-NN) of G are bidirected. Then for each edge (l_i, l_j), if l_i is detected as an attachment to l_j, we retain the directed edge $l_j \rightarrow l_i$ while discarding $l_i \rightarrow l_j$, e.g., $l_a \rightarrow l_b$ and $l_f \rightarrow l_e$ are discarded in Figure 6. The anchoring constraints, however, might conflict with each other, leading to directed cycles[1], where all the edges being oriented in the same direction and each node corresponds to a T-junction (Figure 5c). In such cases, at least one anchoring constraint has to be invalid. Since we have the confidence of each T-junction evaluated, we break directed cycles by removing the anchoring constraint corresponding to the least confident pair (e.g., the highlighted junction in Figure 5c) and change the corresponding edges back to bidirected ones. Although one can let the optimization decide where to break the directed cycles, we found that this may lead to undesirable behavior with strong T-junctions getting removed.

Starting position. The starting location further reduces the computational cost. We observed that simply following the guideline **S1** and starting the drawing process by a line near the topmost of the drawing often leads to unnatural drawing orders. Our input drawings often contain semantically more meaningful objects (e.g., a human figure), where starting with the most important or salient part (e.g., the head) is often desirable. Instead of attempting to infer the starting location from geometry, we alow the user to specify a starting point of interest (e.g., the red dot in Figure 6 (left)) and search for the closest line (e.g., l_b) to this point as the starting line, i.e., as $l_{p_0^*}$. All the directed edges pointing to this line node in the graph are removed (e.g., $l_c \rightarrow l_b$), since no line would be drawn twice. It is possible that the line closest to the user-specified point is an attachment line, which would locally violate the anchoring guideline. We still keep this line as the starting line, since such specific user preference should be respected.

Graph formulation. We associate each graph node l_i with individual line cost $c_{ind}(l_i)$ and each edge (l_i, l_j) with transition cost $c_{tra}(l_i, l_j)$. Our minimization problem now amounts to finding the minimum cost path starting at $l_{p_0^*}$ and visiting each node in G exactly once. This is equivalent to finding a *Hamiltonian path* in our directed graph [Bondy and Murty 1976]. Although the Hamiltonian path problem is NP-complete, given our carefully constructed graph, the problem search space is already significantly smaller than that from a naïve construction. For example, in Figure 4, there are 26 nodes, but instead of $26 \times 25 = 650$ possible directed edges only 145 directed edges are present in the final construction.

Starting from $l_{p_0^*}$, we progressively search by graph connectivity, using a branch-and-bound approach, for finding the Hamiltonian path with the minimum cost, though resorting to other approximations is possible [Bondy and Murty 1976]. During the search, we avoid treating all attachment lines before their corresponding substrate lines, which not only enforces the anchoring constraints but also terminates invalid paths early on.

Choice of k. The value of k in constructing k-NN graph G significantly influences the availability of a Hamiltonian path in G as well as the search computational cost. To avoid combinatorial explosion, we restrict $k \in [3, 8]$ and the number of lines $|L|$ smaller than 35. For a given value of k ($= 4$ by default), possible bidirected edges (not from k-nearest neighborhood) might be added to make G at least weakly connected, i.e., to avoid disconnected components. If no Hamiltonian path exists for the current configuration, we increase the value of k by 1 until we reach the upper threshold. We found this heuristic to work well in practice. Note that when no Hamiltonian path is found, we can simply pick the found longest path with the minimum cost for ordering the involved lines, and sequentially append the rest of the lines simply by their proximity.

Ordering of detail lines. Empirically, we observed that the order of detail strokes is less important and thus we resort to a simpler strategy instead of the computationally expensive strategy as required for the significant lines. The detail lines in a group approximate collection by certain similarity (**R4**) or by semantics. We therefore draw the detail lines group by group, in the spirit of guideline **H2**. The traversal order *across groups* is determined as follows: first the centroid of each group of lines is calculated; then starting from the group containing $l_{p_0^*}$, each group is traversed by proximity based on the distance between the centroids. The drawing order of individual lines *within a group* is also decided by proximity. Unlike significant lines, we found that the choice of the starting line inside each group was less important. Our current system randomly picks a line in a group as the starting line to reflect that details are often included as an afterthought rather than as part of the original plan. This was also confirmed by our user study findings as detailed later.

5.4 Stroke direction determination

Having restored the drawing order of strokes, the remaining task is to determine the directions when drawing individual strokes. We provide the following two schemes based on the guidelines **D1**–**D3**.

Mechanical movement. The first scheme is motivated by the comfortability of mechanical movement of hands and determines the drawing direction of individual strokes independently, **D1** and **D2**. Specifically, the direction is solely determined by the *acute* angle between the line defined by the two end points of the stroke and the x-axis, denoted as α (Figure 7). For right-handers, if α is smaller than some threshold, $\pi/12$ in our case, the stroke will be drawn

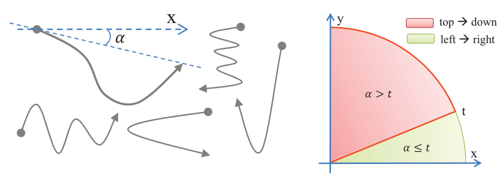


Figure 7: *Illustration for stroke direction by right-handers. Left-handers have a mirrored preference (see supplementary viewer).*

[1]Note that a cycle with at least one bidirected edge is not of our concern.

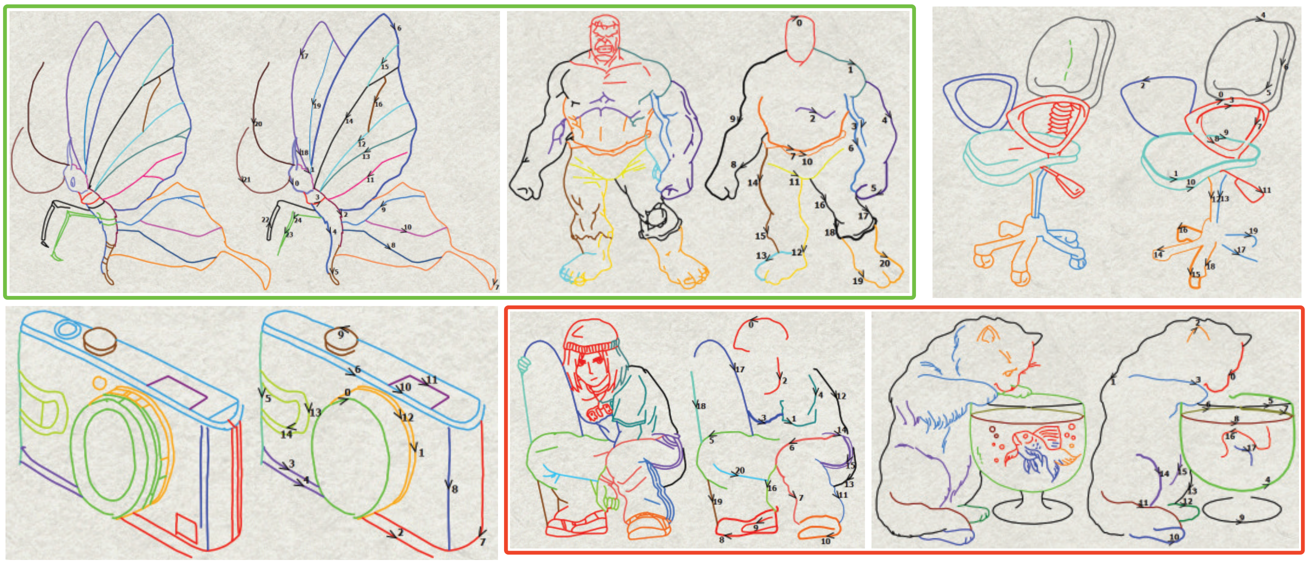


Figure 8: *Results of grouping (groups indicated by colors) and ordering of significant lines (orders by numbers and directions by arrows). Two representative results with the highest and lowest normalized votes in our user study are highlighted in green and red, respectively.*

from left to right. Otherwise, it will be drawn from top to bottom. The stroke directions for left-handers are mirrored.

Economy control. The second scheme is mainly motivated by motor convenience and economy of control when drawing a sequence of lines, **D3**, which is useful for applications like gesture drawing synthesis (see Section 6.2). The stroke directions are influenced by the order of strokes but less so by the handedness of the drawer. The starting point as well as the direction of a stroke is mainly determined by examining which of its two end points is closer to the last drawn line.

6 Results and Discussion

We first discuss the evaluation results obtained via a user study and then a few applications enabled by the derived drawing orders.

6.1 Evaluation

We have tested our algorithm on a wide range of input line drawings, with varying degrees of complexity and structure. Figure 8 shows some examples of grouping and ordering results. See the accompanying video for recorded drawing animations or the animation viewer at the project page for live animations. The computationally expensive step involves finding Hamiltonian paths on k-NN graphs, whose running time ranges from a few seconds to 2-5 minutes, depending on the value of k, the number of significant lines, and the configuration of k-NN graphs. In order to evaluate the visual plausibility of the detected drawing orders as generated by our algorithm, we conducted a user study, which was done in two parts.

Parameters. The algorithm output depends on a few key parameters: number of nearest neighbors k, w in Equation 1, and w_p in Equation 3. Typically parameter values are selected within their corresponding ranges (Section 5). Although in simple examples default values suffice, in complex ones (e.g., Figure 8 camera and boy) manual intervention may be needed (in our examples 2-3 tries). In future, we plan to allow the user to specify a scale of interest and tag semantically important parts.

User study I. First, we evaluate the *relative* effectiveness of our algorithm compared to other possibilities (Figure 11), including random ordering **RO**, longest-first ordering **LFO**, nearest-first ordering **NFO**, and smoothest-first ordering **SFO** (i.e., based on the best collinearity). A random line and the longest line are detected as the starting lines for **RO** and **LFO**, respectively. For **NFO** or **SFO**, we use the same starting line as used by our algorithm. For consistency, we use the scheme of stroke direction determination shown in Figure 7 for all the methods.

In the designed questionnaire (see the project page) together with ordering videos used during the survey, subjects were requested to pick the visually most plausible/reasonable order of a drawing among 5 *anonymous* possibilities from a total of 19 drawing examples (Figure 9 (bottom)). In order to minimize fatigue among the subjects, the examples were split into 4 sets and each subject was shown only 4-5 drawings. Each example was evaluated by more than 20 subjects on average. The participants were in the age range 18-36, 30% females, with most being right-handed. Most of the subjects had no professional drawing training, i.e., have little drawing experience or have learned basic drawing skills only from courses in primary/secondary schools or universities.

Figure 9 (left) show the normalized votes for the five methods. We conducted two-proportion z-tests to determine whether the difference between the normalized votes by a pair of methods for all the examples is statistically significant. Overall our algorithm significantly outperformed the other four methods (p-value < 0.01). Among the local methods, **NFO** and **SFO** worked reasonably well for only certain examples (e.g., Examples 4 and 12 for **NFO** and Example 7 for **SFO**). In particular, **NFO** is the second best method (p-value < 0.01 when comparing it to either **SFO** or **LFO**), conforming to the fact the proximity is the primary influence on drawing order. However, **NFO** typically performs poorly for drawings with a complex mixture of coarse and fine details like Examples 8 and 17. The difference between the normalized votes by **LFO** and **SFO** is not statistically significant (two-tail p-value = 0.728). As expected, **RO** was the least favored.

Typically, humans start by sketching a rough shape and structure, and then introduce details, thus reiterating the importance of building a level-of-detail representation of the input drawing. Among all the examples, our technique gets the lowest votes for Examples 12 and 13. The subjects who prefer **NFO** to ours for Example 12 have predefined semantics-based order for human figure drawing in mind (e.g., head $\rightarrow$ shoulder $\rightarrow$ legs), which by chance

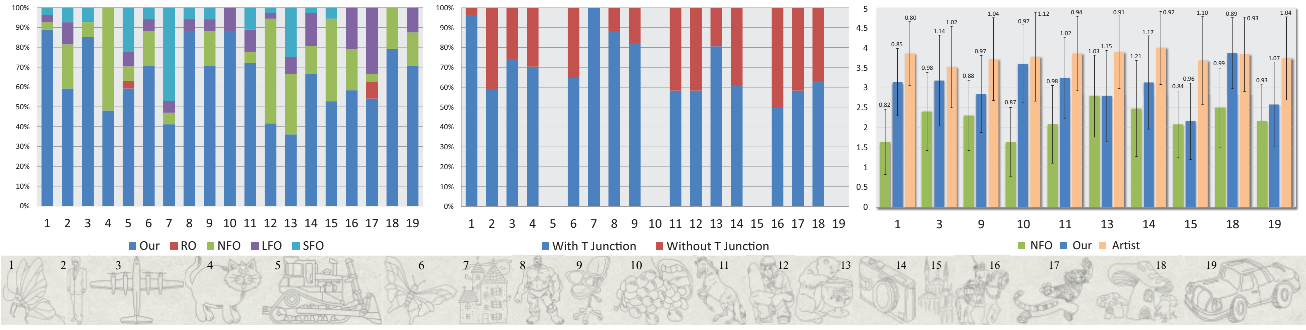


Figure 9: *(Left) Normalized votes for plausibility of results by our technique and the other four straightforward solutions. (Middle) Normalized votes for plausibility of results by our technique with and without T-junction constraints. (Right) The degree of plausibility of orders by* ***NFO****, our technique, and an artist.*

is better captured by **NFO**. The desired order for Example 13 is component-by-component (e.g., cat → cup → goldfish) requiring a level of image understanding, probably unrealistic to be obtained simply from the sketch geometry. Adapting interactive tools designed for cartoon processing might be helpful to interactively encode component-level orders (see Sýkora et al. [2009; 2010]).

We also evaluated the effectiveness of introducing T-junction constraints (Figure 9 (middle)) by turning graph G back to an undirected graph to test the effect of ignoring T-junctions. As expected, the orderings with T-junctions are visually more plausible than orderings that ignore such constraints, verified by a two-proportion z-test with p-value < 0.01. T-junctions are especially useful for drawings like Examples 1 and 7, where the attachment of lines to the substrate lines is abundant and visually salient. Introducing T-junction constraints also significantly reduce time complexity and memory consumption. Note that Examples 5, 10, 15, and 19 are absent in the comparison due to memory restrictions (4G RAM in our case) for our technique without T-junction constraints.

User study II. Next, we evaluated the degree of plausibility of the estimated orders to human viewers. Only the drawings for which we have the corresponding original orders employed by an artist are included (Figure 9 (right)) such that the plausibility of the orders by the artist can also be evaluated for reference. We deliberately avoid using the set of lines employed by the artist as input to our algorithm, since in most cases we have to rely on automatic vectorization and tolerate errors from this step. The orders by **NFO**, which was the most favored among the simple alternatives, are included as well. Note that in highly structured scenes with many non-local part relations, plausibility may be partially dependent on target applications, e.g., specialized instructional animation. However, in this work, we ignore such secondary effects.

Participants were requested to rate the plausibility of *individual* orders on a discrete scale from 1 to 5, where 1 means completely unreasonable, 3 plausible and 5 perfectly plausible (done by a human being). Figure 9 (right) plots the mean and the corresponding standard deviation for individual orders, based on the feedback from 40 subjects in age range 18–41 (around 30% with artistic training and 35% being female) for each example. Since how well humans can judge the degree of plausibility is unclear, the exact meaning of rating scores here might not be very important. Below we show that the results by **NFO** and the artist would serve as a lower bound and upper bound, respectively.

We performed a paired t-test on the scores between our method and **NFO**. We found a significant difference (p-values < 0.05) between these two methods for most of the models. A paired t-test also verified that our method is *overall* more effective than **NFO** (M_{our} = 3.07, M_{NFO} = 2.18, p-value = 6.6E-31 with $\alpha = 0.01$). This is consistent with our conclusion for comparing relative performance of the different ordering techniques (Figure 9 (left)). Since we only consider geometric cues, it is expected that our results would not be better than those by the artist, verified by a paired t-test between the scores by our method for all the examples and the corresponding scores for the artist (p-value < 0.01). Surprisingly, however, in many cases, participants had difficulty distinguishing between human ordering and our algorithm output. For example, the results between our method and the artist for Examples 10 and 18 are not significantly different (p-value > 0.05). It is interesting to note that the artist's results are moderately underrated (with mean as 3.78), reiterating that drawing order often involves personal preferences.

6.2 Applications

Animation. Animation is often more powerful than static drawing. By drawing lines one by one in the inferred order and direction, we can easily produce an animation simulating the drawing process of the original image. We show that vivid drawing animations can be produced by augmenting the animations with secondary sound effects and various drawing tools, e.g., a pencil (Figures 1 and 10). Since drawing speed drops when going around end points and high-curvature points of curves [Sezgin et al. 2001], we slow down the drawing speed at such points (see supplementary video).

We show the utility of synthesized drawing animations in two applications. First, it is beneficial for expressing images whose content may be perceived differently under different conditions, e.g., famous optical illusion figure of Rubin's vase (Figure 10). Our animation is able to express such content organization in those images. To achieve this, the user scribbles a segmentation to partition the set of input lines to a few components (Figures 10 and 12) and assign the drawing order of components. The lines within each component are then ordered by our algorithm (Section 5). Another application is for video scribing, i.e., storytelling via drawing animation with narration, which is a popular multimedia form nowadays (e.g., see PhD comics #1430, and RSA Animate). Instead of recording drawing animations for specific stories or topics, our technique allows easy generation of drawing animation of static scenes by manual segmentation of components, assignment of component order with respect to the narration, and interactive camera planning for properly conveying context information (based on region of interests

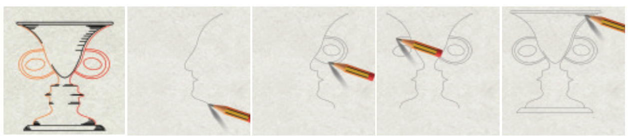


Figure 10: *Animation of faces-vase drawing. (Left) Manually segmented components, indicated by color.*

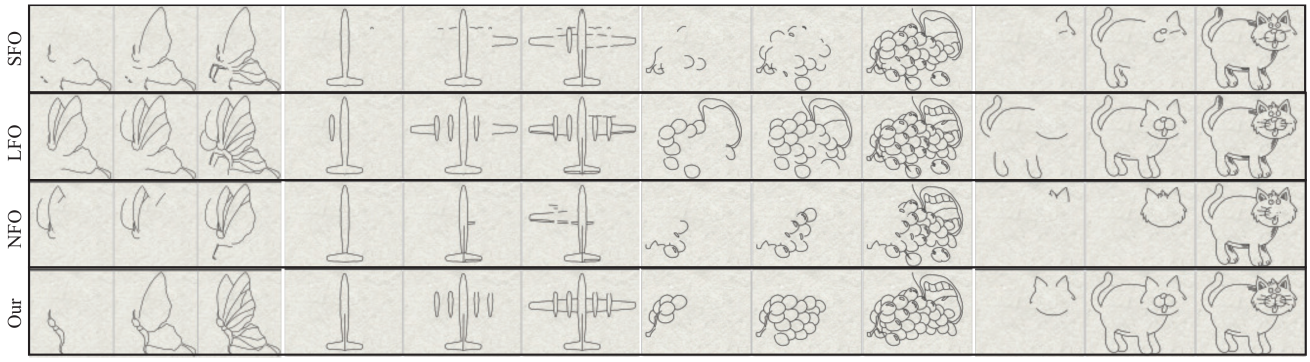


Figure 11: *Screen shots after drawing the same numbers of lines with different methods.*

according to coarse level lines). This opens the door for reusing vast amount of line drawings on the Internet by combining them for different stories (see Figure 12, supplementary video and demo viewer).

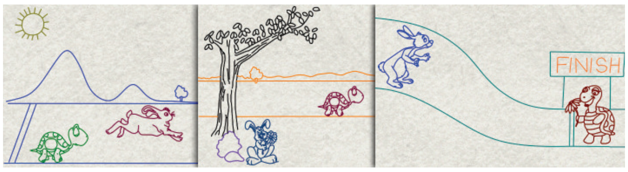


Figure 12: *Animation for vivid storytelling. See the accompanying video for the narrated animation. Components by manual segmentation are indicated by color.*

Gesture drawing. The estimated order provides temporal information for every stroke in the drawing. Such temporal information offers the possibility for "temporal simplification" of the drawing, i.e., what would it be like if the drawing were made in a much shorter amount of time? This motivates us to synthesize a gesture drawing effect from the input drawing (Figure 13). A gesture drawing is a work of art defined by rapid execution [Nicolaides 1990]. Its purpose is to help a drawer record her impressions, thoughts and feelings about a subject in a short amount of time (typically less than 2 minutes). Therefore, in gesture drawing, a pencil keeps drawing rapidly and continuously in a ceaseless line, without taking the pencil off the paper. To achieve this for an ordered set of lines with the second scheme for stroke direction, we randomly link temporally adjacent lines in a time window (of 5 in our case) and then replace the original lines with Bézier curves that interpolate corresponding endpoints and associated tangents of the linked lines.

Limitations. Our current implementation does not explicitly detect and encode similar lines or symmetry patterns, though the order of such lines (e.g., in Examples 3 and 6) already appears reasonable due to the anchoring and collinearity guidelines. It is possible to explicitly encode similarity and symmetry into the ordering process by representing sets of similar or symmetry lines as single graph nodes, on which our current energy terms can be similarly defined. See the supplemental viewer for our preliminary experiment on examples (see inset) with manually labeled reflective symmetry, which would be interesting to explore further as a future work. Currently, we assume the inputs to be relatively clean, free of hatching strokes. Similarity among local detail strokes may help in automatically detecting and handling such stylized sketches. Lastly, we still do not have a quantitative understanding about algorithmic limits or ambiguities.

7 Conclusion

We introduced the problem of animated construction of line art, and summarized the key principles of drawing order from cognition literature. We first construct a hierarchical grouping of the input sketch curves, and then traverse the resultant hierarchy in an order dictated by an energy formulation based on computational realizations of the drawing order guidelines. We obtain the ordering using an approximate Hamiltonian path on an appropriately constructed graph. We tested our framework on a range of input sketches, and evaluated the results via a user study. Lastly, we demonstrated potential applications especially for creating video scribing.

Future Work. Interesting research challenges lie ahead. While we continue to explore better geometric sketch analysis techniques, other attributes may also carry important cues. For instance, some line art images, e.g., synthesized from natural images, are often equipped with color information for regions enclosed by lines. Such color information may contain hints for grouping of lines and help in semantic cues for hierarchy construction, greatly influencing the ordering of lines. Another example is line art images generated by rendering 3D models using line drawing rendering techniques [Grabli et al. 2010]. The corresponding depth, occlusion, and segmentation information extracted from the original 3D models can help resolve ambiguity in line ordering. Although our framework allows easy integration of extra ordering cues, including thickness, color and depth, further research needs to be done to judge the specific benefits. As an application, the inferred (plausible) drawing order can be directly highlighted and stylized using adaptive rendering styles [House and Singh 2007]. Similarly modulating other attributes like line width and color can also produce interesting sketch stylization.

Acknowledgements. We thank the reviewers for their constructive comments, the many participants of the user studies for their time, Gail Redman for video narration, Kanlun Cen for the line drawings used in this paper, and Wilmot Li for his many insightful comments. Hongbo Fu was supported by grants from CityU (No. SRG7002533), and the HKSAR Research Grants Council (No. 9041562); Ligang Liu was supported by the National Natu-

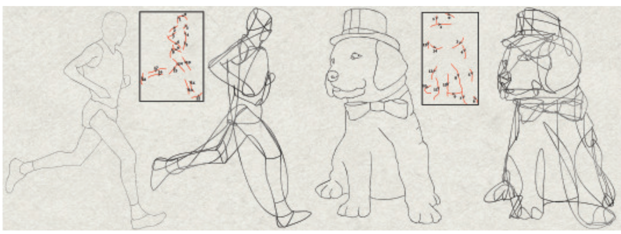


Figure 13: *Gesture drawing synthesis.*

ral Science Foundation of China (61070071) and the 973 National Key Basic Research Foundation of China (2009CB320801); Niloy J. Mitra was partially supported by KAUST research network.

References

BABCOCK, M. K., AND FREYD, J. J. 1988. Perception of dynamic information in static handwritten forms. *The American Journal of Psychology 101*, 1, 111–130.

BARLA, P., THOLLOT, J., AND SILLION, F. 2005. Geometric clustering for line drawing simplification. In *EGSR*, 183–192.

BONDY, J., AND MURTY, U. 1976. *Graph Theory with Applications*, vol. 290. MacMillan London.

COHEN, H. 1995. The further exploits of AARON, painter. *Stanford Humanities Review 4*, 2, 144–160.

COLE, F., GOLOVINSKIY, A., LIMPAECHER, A., BARROS, H. S., FINKELSTEIN, A., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2008. Where do people draw lines? *ACM Transactions on Graphics 27*, 88:1–88:11.

DECARLO, D., AND STONE, M. 2010. Visual explanations. In *NPAR*, 173–178.

DODSON, B. 1990. *Keys to Drawing*. North Light Books.

DURAND, F. 2002. An invitation to discuss computer depiction. In *NPAR*, 111–124.

EDWARDS, B. 1999. *The New Drawing on the Right Side of the Brain*. Jeremy P. Tarcher/Putnam New York.

GOMBRICH, E. H. 1960. *Art and Illusion: A Study in the Psychology of Pictorial Representation*. Phaidon Press.

GOOCH, B., AND GOOCH, A. 2001. *Non-Photorealistic Rendering*. AK Peters, Ltd.

GRABLI, S., DURAND, F., AND SILLION, F. X. 2004. Density measure for line-drawing simplification. In *PG '04*, 309–318.

GRABLI, S., TURQUIN, E., DURAND, F., AND SILLION, F. X. 2010. Programmable rendering of line drawing from 3d scenes. *ACM Transactions on Graphics 29*, 2, Article 18.

GUPTILL, A., AND MEYER, S. 1997. *Rendering in Pen and Ink*. Watson-Guptill.

HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH '98*, 453–460.

HOUSE, D. H., AND SINGH, M. 2007. Line drawing as a dynamic process. In *Pacific Graphics '07*, 351–360.

JUDD, T., DURANT, F., AND ADELSON, T. 2007. Apparent ridges for line drawing. *ACM Transactions on Graphics 26*, Article 19.

KATO, Y., AND YASUHARA, M. 1999. Recovery of drawing order from scanned images of multi-stroke handwriting. In *ICDAR '99*, 261–264.

LEE, Y., MARKOSIAN, L., LEE, S., AND HUGHES, J. F. 2007. Line drawings via abstracted shading. *ACM Transactions on Graphics 26*, 3, Article 18.

LI, G., LIU, L., ZHENG, H., AND MITRA, N. J. 2010. Analysis, reconstruction and manipulation using arterial snakes. *ACM Transactions on Graphics 29*, 6, 152:1–152:10.

MEHRA, R., ZHOU, Q., LONG, J., SHEFFER, A., GOOCH, A., AND MITRA, N. J. 2009. Abstraction of man-made shapes. *ACM Transactions on Graphics 28*, 5, 137:1–137:10.

NGUYEN, V., AND BLUMENSTEIN, M. 2010. Techniques for static handwriting trajectory recovery: a survey. In *International Workshop on Document Analysis Systems*, 463–470.

NICOLAIDES, K. 1990. *The Natural Way to Draw: a Working Plan for Art Study*. Mariner Books.

NORTHAM, L., ISTEAD, J., AND KAPLAN, C. 2010. Brush stroke ordering techniques for painterly rendering. In *Computational Aesthetics in Graphics, Visualization, and Imaging*.

NOVICK, L., AND TVERSKY, B. 1987. Cognitive constraints on ordering operations: The case of geometric analogies. *Journal of Experimental Psychology: General 116*, 1, 50–67.

NOVICK, L., AND TVERSKY, B. 1999. What does drawing reveal about thinking? *Visual and spatial reasining in design*, 93–101.

ROSIN, P. 2002. Techniques for assessing polygonal approximations of curves. *IEEE TPAMI 19*, 6, 659–666.

SEZGIN, T. M., STAHOVICH, T., AND DAVIS, R. 2001. Sketch based interfaces: Early processing for sketch understanding. In *Proceedings of PUI-2001*.

SHESH, A., AND CHEN, B. 2008. Efficient and dynamic simplification of line drawings. *Computer Graphics Forum 27*, 2, 537–545.

STROTHOTTE, T., AND SCHLECHTWEG, S. 2002. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufmann Pub.

SÝKORA, D., DINGLIANA, J., AND COLLINS, S. 2009. Lazybrush: Flexible painting tool for hand-drawn cartoons. *Computer Graphics Forum 28*, 2, 599–608.

SÝKORA, D., SEDLACEK, D., JINCHAO, S., DINGLIANA, J., AND COLLINS, S. 2010. Adding depth to cartoons using sparse depth (in)equalities. *Computer Graphics Forum 29*, 2, 615–623.

TOMBRE, K. 1998. Analysis of engineering drawings: State of the art and challenges. *Graphics Recognition Algorithms and Systems*, 257–264.

TVERSKY, B., AND SUWA, M. 2009. Thinking with sketches. *Tools for Innovation 1*, 9, 75–85.

VAN SOMMERS, P. 1984. *Drawing and Cognition: Descriptive and Experimental Studies of Graphic Production Processes*. Cambridge University Press Cambridge, UK.

VIVIANI, P., AND STUCCHI, N. 1992. Biological movements look uniform: Evidence of motor-perceptual interactions. *Journal of Experimental Psychology: Human Perception and Performance 18*, 3, 603–623.

WILLATS, J. 1997. *Art and Representation: New Principles in the Analysis of Pictures*. Princeton Univ Pr.

WILSON, B., AND MA, K.-L. 2004. Rendering complexity in computer-generated pen-and-ink illustrations. In *NPAR*, 129–137.

WINKENBACH, G., AND SALESIN, D. 1994. Computer-generated pen-and-ink illustration. In *SIGGRAPH '94*, 91–100.

WU, Y.-F., ZHUANG, Y.-T., PAN, Y.-H., AND WU, F. 2007. Using condensation tracking to recover stroke order of chinese calligraphic handwritings with ccm. In *International Workshop on Image Analysis for Multimedia Interactive Services*.