Dynamic Sketching: Simulating the Process of Observational Drawing

Jingbo Liu[1] Hongbo Fu[2] Chiew-Lan Tai[1]
[1]Hong Kong University of Science and Technology [2]City University of Hong Kong

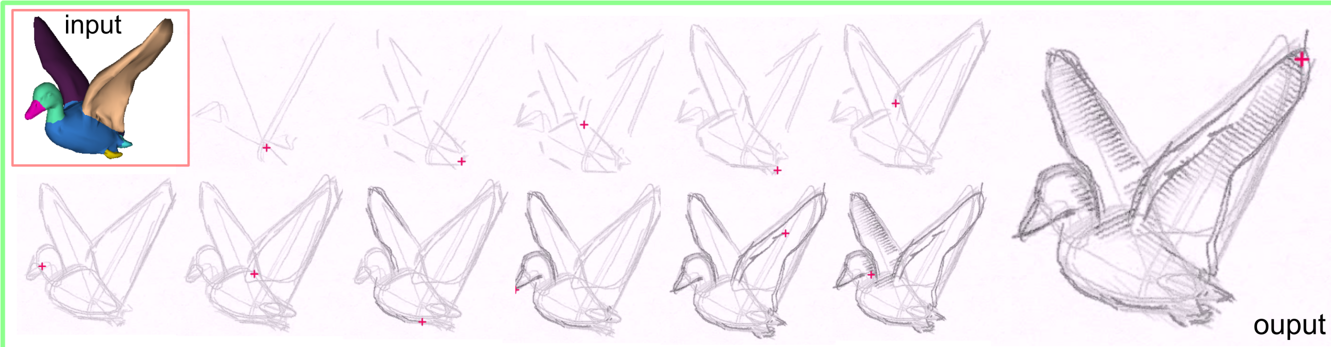


Figure 1: *Our technique simulates the process of drawing by observation. Given a 3D model with semantic segmentation and the current viewpoint, our technique automatically simulates a visually plausible dynamic sketching process.*

Abstract

The creation process of a drawing provides a vivid visual progression, allowing the audience to better comprehend the drawing. It also enables numerous stroke-based rendering techniques. In this work we tackle the problem of simulating the process of observational drawing, that is, *how* people draw lines when sketching a given 3D model. We present a multi-phase drawing framework and the concept of sketching entropy, which provides a unified way to model stroke selection and ordering, both within and across phases. We demonstrate the proposed ideas for the sketching of organic objects and show a visually plausible simulation of their dynamic sketching process.

CR Categories: I.3.3 [Computer Graphics]—Line and curve generation J.5 [Arts and Humanities]: Fine arts;

Keywords: observational drawing, sketching, animation

1 Introduction

Art is more a creative journey than a deliverable end product. The creation process of a drawing, which captures the visual progression of the drawing by an artist, is valuable, since it shows how the artist expresses emotions and artistic apprehension, allowing the audience to better appreciate the drawing. See the rich "speed drawing" examples on video-sharing websites like Youtube. A stroke-by-stroke drawing animation also has its educational value to the viewer in seeing the drawing process unfold. It also benefits applications like visual storytelling using hand-drawn animation (also known as video scribing) [Fu et al. 2011], and enables new stroke-based non-photorealistic rendering effects.

Motivated by the above applications, we aim to simulate the process of observational drawing, that is, how an artist progressively sketches a given 3D object starting from a blank canvas to the final drawing. We focus on the sketching of organic objects, though an extension to man-made objects is also briefly discussed. Given a static 3D model with semantic segmentation, we intend to automatically produce a *dynamic sketching* animation (see the accompanying video and Figure 1) that looks like it is done by a certain novice artist. Our goal is thus very different from those of a large body of existing works on synthesizing visually compelling *line drawing images* from 3D models [Cole et al. 2008].

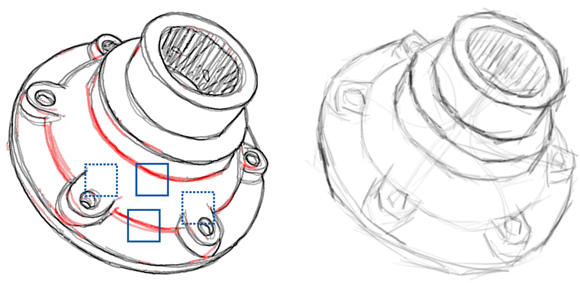
Our technique is inspired by common drawing processes from drawing books, such as *Force* [Mattesi 2011], as well as findings from cognitive science: even though the sketching process may be distinctive among artists, it generally progresses in a similar fashion, from coarse to fine, global to local, near to far, and inconspicuous to salient parts [Mattesi 2011]. We focus on simulating the most common and basic phases of the drawing process and present a multi-phase sketching process, each phase abstracting the subject at a different level of details. We introduce the concept of sketching entropy, enabling a novel selection-cum-ordering approach that favors maximizing the shape information during the drawing process, either in the individual phases or across different phases.

We conduct a user study to evaluate the visual plausibility of the simulated drawing processes and the effectiveness of our proposed method. Our experiment confirms that our results are visually plausible. A statistical analysis shows that our entropy-based ordering strategy leads to more plausible results than those driven by the conventional Gestalt rules used in previous work [Fu et al. 2011].

2 Related Work

Synthesizing line drawings from static or dynamic 3D models has been extensively studied. A detailed review on this topic is beyond the scope of this paper. Please refer to [Rusinkiewicz et al. 2008] for an insightful survey. Many techniques have been proposed to simulate *where* people draw lines for a given 3D model [Cole et al. 2008]. As their target is essentially a stylized visualization of the input model, they focus on finding lines lying either on the 3D model or its projected image. In contrast, the human drawing process typically starts with guiding lines or structure lines, for example, pos-

CAe 2014, August 08 – 10, 2014, Vancouver, British Columbia, Canada.



(a) *Artists' drawing* **(b)** *Our result*

Figure 2: *(a) Certain lines (highlighted in red) are commonly drawn by artists but difficult to synthesize with the traditional line drawing synthesis techniques [Cole et al. 2008]. (b) Our technique is able to find such lines.*

ture lines for figure drawing [Guay et al. 2013], which are not necessarily on the surface of the model (Figure 2). The synthesis of such guiding lines has been limitedly explored in [Grabli et al. 2010]. Their technique again focuses on producing non-photorealistic renderings of 3D models, rather than the temporal aspect of the drawing process.

The work proposed by Fu et al. [2011] is probably the most relevant to ours. By mapping the Gestalt rules to computational procedures, their technique is able to generate a human-like drawing animation from a given image of line drawing. The output video is visually plausible for storytelling, and the generated process is close to how a human would draw in reproducing an existing line drawing. However, it considers only lines that appear in the given line drawing. It is not clear how to integrate common sketching behaviors, such as refinement and retracing, into their framework. We will show that a straightforward extension of their ordering technique based on the Gestalt rules produces less plausible results than our entropy-based method.

3 Methodology

The sketching process commonly varies among people or even for the same person in different times. We thus aim to produce a sketching animation that looks visually plausible to human viewers, instead of the sketching process of a specific artist. To achieve this goal, we base our solution on findings from cognitive science. In this work we focus on drawing organic objects only.

3.1 Drawing and Cognition

Cognitive science reveals that what geometric elements artists draw and their drawing order reflect how artists schematize and conceptualize objects [Tversky and Suwa 2009]. In the early stages of drawing, the cognition process requires conceptual skills, but requires perceptual skills in the later stages [Suwa 2003]. With conceptual skills, artists analyze the drawing subject and abstract it into meaningful simpler elements that make up the whole object. As drawing progresses, the artists then treat the drawing subject as a whole and focus on adding perceptual features that were not expressed in the conceptual stage.

The drawing process of humans comprises a series of actions, presented structurally and hierarchically [Tversky et al. 2006]. The actions of a common sketching hierarchy can firstly be divided in the temporal domain, forming different *phases* of drawing. Each phase abstracts the subject at a certain level of details, and refines the previous phase by adding more details. Abstraction rather than simplification plays a critical role [Kavakli et al. 1998]. The actions in each phase can be further divided in the spatial domain, where each set of strokes refines a specific object part. In other words, artists mentally decompose the drawing subject by perceptual or functional salience, and organize the drawing process in each phase by object parts.

3.2 Sketching Phases

To capture common characteristics of the way humans draw, we propose to simulate the drawing process in a sequence of phases. The drawing actions are ordered *phase by phase*, since drawings, being reflections of our mental abstraction, have a natural multi-layered structure [Novick and Tversky 1987].

A different type of object might need a different series of drawing phases. For example, the way an artist draws a man-object object like a chair is often different from that of drawing an animal [Team 2005]. Below we present a typical drawing process for organic models, consisting of four phases (Figure 3): *posture phase*, *primitive phase*, *contour phase*, and finally *detail phase*. Again it is worth reiterating that the entire sequence of phases might not be adopted by every artist or every object, and an artist might not exactly follow the phase-by-phase order. However, we expect that the resulting animation would have a reasonably high degree of visual plausibility, which is essentially our goal.

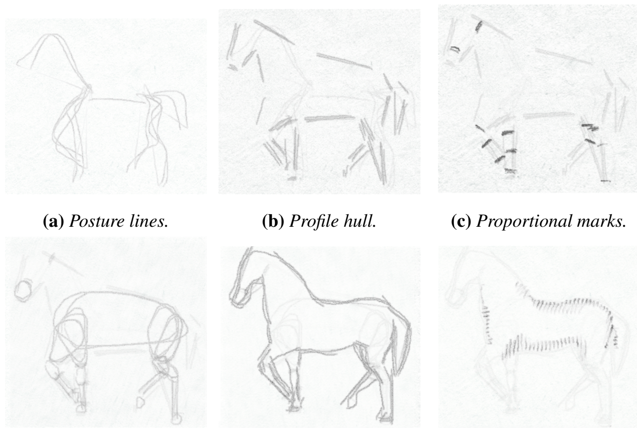
Our posture phase and primitive phase simulate how artists use conceptual skills to sketch an object. The posture phase is the first stage where artists express their understanding of an organic object. Artists draw a very small number of lines to capture the motion or posture of the subject [Guay et al. 2013]. In the primitive phase, lines are often used for rough space allocation for individual parts. Similar lines have recently been used as visual guides to support drawing [Iarussi et al. 2013].

To enable such phasing, we require a semantic decomposition of the input 3D model (Figure 1) to be available to our main algorithm. Semantic mesh segmentation is a well-studied problem and is not our focus. In our implementation, we resort to a simple interactive segmentation system [Zheng et al. 2012]. After segmentation, intersegment analysis is performed to detect symmetric parts or parts of similar shape [Zheng et al. 2013].

Next we describe how we identify the strokes in each phase. These strokes will be turned into humanized trajectories to emulate the drawing process (see section 3.3).

Posture Phase. In this first phase our system draws posture lines (Figure 3a), which convey the layout and relative relationships among object parts. Posture lines are like the armature wires used to hold the clay in sculptures and are sketched like the way a sculptor bends the armature wires. These lines are not about accuracy or likeness but to capture the pose and action. They serve to guide the subsequent strokes that populate the subject's body; the likeness will be built by other lines in subsequent phases [Mattesi 2011]. We model these lines as a 3D curve-skeleton of the model, which can be extracted from the segmentation information [Lien et al. 2006]. The extracted skeleton is a connected 3D curve network. Splitting the skeleton at junctions leads to individual posture lines. As artists tend to draw straighter and smoother lines in earlier phases, we first simplify the posture lines using a simple line approximation algorithm and then fit them with B-spline curves. The resulting posture lines roughly conform to C- or S-shaped curves to produce lines of actions [Guay et al. 2013].

Primitive Phase. In this phase, the drawing is refined by clearer allocation of the space for individual object parts. Our system starts



(a) *Posture lines.* (b) *Profile hull.* (c) *Proportional marks.*

(d) *Ovals.* (e) *Contour lines.* (f) *Hatching.*

Figure 3: *Simulating a typical drawing procedure for organic objects. (a) Posture phase. (b-d): Primitive phase. (e) Contour phase. (f) Detail phase.*

with drawing a *profile hull*, which roughly captures the space of one or more object parts (Figure 3b). The profile hull is computed as a greatly simplified silhouette of the 3D model.

Artists often develop volumes for object parts using simple geometric primitives [Tversky and Suwa 2009]. As artists do, we analyze the given 3D shape to construct simple primitives forming the whole object. Our system first draws proportional marks on the posture lines to delimit the spaces to be filled with primitives. As shown in Figure 3c, such marks orthogonally cross the posture lines and divide the postures lines into portions.

Next our system draws primitives for object parts, *part by part*. For organic shapes, a natural choice of primitives is deformed ellipsoids. However, since the projection of ellipsoids essentially leads to 2D oval shapes, we directly draw 2D oval-like shapes, which is undoubtedly a most popular sketching element used in observational drawing of organic objects. The oval-like shapes are not necessarily symmetric or convex (see Figure 3d). Not all object parts properly fit ovals (e.g., neck and legs of horse in Figure 3d). For such cases, our system draws a simple hull by simplifying the silhouette of the object part instead of the fitted oval.

Contour Phase. The contour phase and detail phase constitute the perceptual stage. In the contour phase, the parts are integrated into a complete object (Figure 3e). To construct the perceptual shape, the conceptual primitives are retained, and the unexpressed contours are drawn first before the others that just refine existing lines. Specifically, contour lines that either connect the drawn primitives to form a complete silhouette or modify primitive lines that are farther from the shape contour are drawn first. To achieve this effect, we first extract all the suggestive contours and highlight chains [DeCarlo and Rusinkiewicz 2007] and break them at corners of large concavity, resulting in several excessive sets of suggestive contour segments. Our entropy-based selection and ordering algorithm (Section 3.4) then automatically draws all the unexpressed contour segments first, followed by the rest of the lines.

Detail Phase. The perceptual saliency is fully revealed in this phase. This final stage of the sketching process briefly shades the subject with hatching (Figure 3f). Hatching is one more step towards the perceptual view: it adds a sense of substance to the drawing subject and produces a convincing tonal relationship, which then translates to a sense of depth. Hatching can also take place across primitives, which further integrates object parts into one subject. We implement a simple method to produce such hatching strokes in real time, using the top influential properties suggested in Kalogerakis et al. [2012].

Options. We provide a simple user interface so that the user may disable one or even more phases to get customized results. For example, an experienced artist might omit the primitive phase, the hatching effect is not a must in every line drawing process, and the posture lines are usually not needed for the sketching of man-made objects (e.g., the twoboxcloth model in Figure 8). To simulate the sketching process of man-made objects, we may extend our current drawing phases by, for example, replacing skeletons with symmetry axes, and replacing ovals with 3D primitives such as cuboids, generalized cylinders and ellipsoids.

3.3 Stroke Rendering

Humanized Trajectory. We use a variant of the Proportional Integral Derivative (PID) system by House and Singh [2007] to obtain a non-photorealistic rendering of the lines in all the phases. The PID system is a control system that simulates the action of a human moving a pen, to produce a drawing process similar to that of an artist during observational drawing. The original PID system is capable of generating human-like expressive strokes, but suffers doodling artifacts at small damping ratios or large pen masses, thus failing to obtain compelling loose strokes.

To eliminate such artifacts, we enhance the PID system with two masses. Connected with a spring and a damper, one mass represents the pen and the other represents the tracker on the target trajectory, respectively. The tracker's mass is then set as a small value proportional to the damping ratio. In this way the tracker always follows the pen closely, and generates strokes without the doodling artifacts. The enhanced system is able to produce expressive strokes at any range of damping ratio, with a wide variety of stroke freedom. Please see the detailed description of the enhanced control system in the appendix. Specifically, our control system is able to simulate the quicker and looser strokes drawn in the earlier phases, and produce the retracing effect of both open and closed lines. The retracing starts relatively loosely and rapidly, but gradually becomes slower and closer to the target shape as the retracing continues (see the accompanying video). Such a wide range of stroke freedom reproduces the dynamics of the drawing process.

Intensity, Thickness and Speed. A monotonic stroke with constant intensity, thickness and speed is dull. Artists use darker and thicker lines to emphasize important features. Their strokes move faster for rougher lines, but slower when depicting details. Based on these observations we set the rendering properties with respect to the point-wise information of each stroke. The point-wise information is computed as the entropy information (defined in Section 3.4) conveyed by the tangent vector at every point along the stroke. This introduces more dynamics to our animation.

3.4 Stroke Ordering

To reflect the hierarchical nature of the human drawing process, we propose to order the lines in the four phases at three levels: first *phase by phase*, then *part by part*, and finally *stroke by stroke*. The phase-level order is always fixed: from the posture phase to the detail phase, though some of the phases may be interactively omitted. In this section we discuss how to achieve the part-level orders in individual phases and the stroke-level orders in individual parts.

In the posture and primitive phases, a small number of construction strokes are drawn to depict the global layout and space allocation of individual parts. Artists' real drawing processes reveal that strokes that bring more information to the current drawing, typically those that are farther from existing strokes, are often drawn first. See such an illustration in Figure 4e–h. In contrast, the Gestalt rules, where proximity plays a dominant role in determining stroke ordering [Fu et al. 2011], often result in a growing process (Figure 4a–d). A similar behavior happens in the contour phase: artists tend to first draw lines that are not expressed in the earlier phases. These observations motivated us to model the sketching process using an entropy model, where each new stroke is intended to maximize the information in the developing drawing [Shannon 1948]. The sketching entropy provides a unified way to model the selection of strokes and their ordering across all phases.

Sketching Entropy. Essentially, we want to measure the information gain of a stroke s given the set of previously drawn strokes C^s before s, against the target drawing as the ground truth. For simplicity, we use the rendered mesh of the input model as the target drawing (e.g. In Figure 5 left). Let $\{v_i\}$ denote the whole set of salient features in the target drawing, for example, represented as normalized tangent vectors along the iso-contours of the target drawing's illumination, defined for every pixel in the drawing. Assuming that the events of strokes capturing such features are independent, the probability that the stroke s given C^s will appear in the drawing, $p(s|C^s)$, is computed with regard to each individual feature:

$$p(s|C^s) = \prod_i p_s(v_i|C^s), \quad (1)$$

where $p_s(v_i|C^s)$ is the probability of having at least one arbitrary stroke closer to v_i than s. Intuitively, a larger probability value means s is farther from v_i.

We model the distance $F(t, u)$ from a stroke t to a feature u as a Poisson process. By the Poission assumption,

$$p_s(v) = -log(1 - e^{-\lambda(F(v,v)-F(s,v))}), \quad (2)$$

where $F(t, u)$ measures both Euclidean and angular distances between t and u. Therefore, $p_s(v_i|C^s)$ can be computed as $P(N[F(s, v_i)] - N[F(v_i, v_i)] \geq 1|C^s)$, where N is the counting function of the Poisson process. See the appendix for the detailed formulation of $F(t, u)$ and $p_s(v_i|C^s)$. If s captures more of a feature v_i, namely, s and v_i are closer, the probability of s appearing with regard to feature v_i is smaller. The extra *shape information* that is brought by s but not expressed by C^s can be computed as a conditional information gain as

$$I(s|C^s) = -\log(p(s|C^s) = -\sum_i \log p_s(v_i|C^s). \quad (3)$$

Clearly, the information provided by s is diminished by C^s.

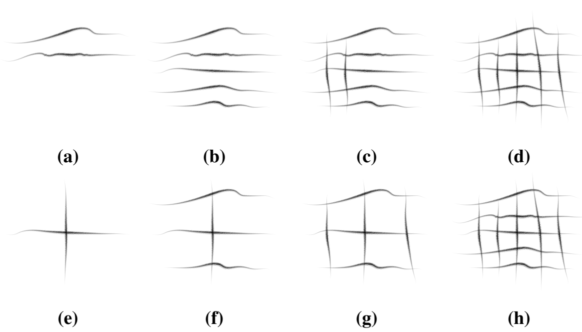


Figure 4: *Illustration of stroke ordering driven by parallel and proximity Gestalt rules (top) and sketching entropy (bottom).*

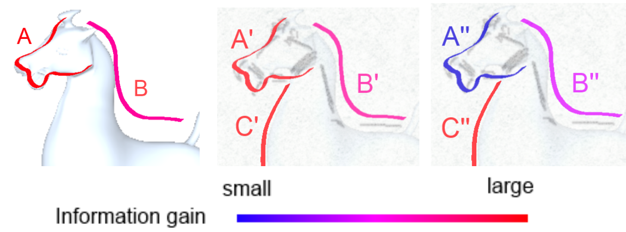


Figure 5: *Left: the information gain of a stroke is generally larger when it is closer to the target drawing. Each stroke is color coded according to its associated information. Middle: unconditional information gain, i.e., by calculating $I(s)$ without considering the previously drawn strokes (C^s) (those in pencil texture). Right: conditional gain $I(s|C^s)$.*

Figure 5 (middle) and (right) illustrate the information gain of a stroke using the unconditional $I(s)$ and conditional formulations $I(s|C^s)$, respectively. Observe that we always have $I(s|C^s) \leq I(s)$, since for $I(s|C^s)$ the previous drawn strokes diminish an upcoming stroke's newly conveyed information. For regions that have not been covered by C^s, the unconditional and conditional information gains are largely the same (strokes C' and C"). When C^s are drawn roughly, the gain of the new stroke is slightly reduced (see the information gain of strokes B' and B"). On the other hand, when s goes to the regions where the previous strokes are rather accurate and complete, the conditional gain of s will be very small (see strokes A' and A"). In an extreme case where s is fully covered by the previously drawn strokes, s conveys zero information.

Entropy-based Selection. As discussed above, the conditional information gain of a stroke could be zero. Such a stroke conveys no new information, either because the features conveyed by this stroke are already present, or the stroke is an outlier. Such redundant strokes are then discarded and not drawn. That is, the conditional information gain also serves to detect and avoid clutters and outliers. The sketching entropy enables the selection of *informative* strokes in addition to ordering. In contrast, a Gestalt-driven ordering strategy draws all input strokes.

Our entropy-based selection bears some resemblance to density measure proposed for line-drawing simplification [Grabli et al. 2004]. The key difference is that density measure does not prioritize the strokes. The line omission depends on an order of input strokes, which is given manually using existing tools. In contrast, as we will describe below, sketching entropy can be used to derive the order of strokes.

Entropy-based Ordering. We model our ordering problem as finding a suitable path in a clustered graph. We let a clustered node in the graph represent a set of strokes within a mesh segment for part-level ordering, and let a primitive node simply represent a single stroke for stroke-level ordering. Each graph edge encodes the relationships between a pair of nodes, for both clustered and primitive nodes. In our current implementation, we encode only Gestalt relations including proximity, parallel and symmetry. We assign a value to each edge according to a prescribed priority of a Gestalt rule, similar to [Fu et al. 2011]. Smaller values mean higher priority.

Now we describe how we find the ordering of nodes within a phase. The same idea applies to both part-level ordering and stroke-level ordering. Let a and b be two graph nodes, with $E(< a, b >)$ representing the value associated to the edge connecting a and b. We

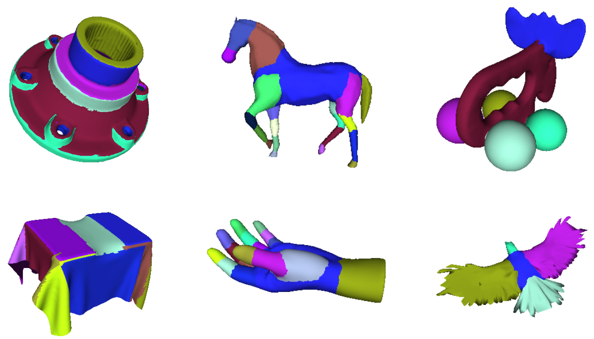


Figure 6: *Input segmented models of the sketching snapshots shown in Figure 2 and 8.*

define the cost of drawing b after a by

$$f(a,b) = ||(-I(b|C^b)) + w_1 E(<a,b>)||, \quad (4)$$

where $I(b|C^b)$ is the conditional information gain contained in node b, given a set of previously drawn strokes C^b, $E(<a,b>)$ measures the contribution due to the Gestalt rules, and w_1 is a balancing weight. We let $w_1 = 0$ for the posture, primitive and detail phases so that our sketching entropy completely determines the order of the involved strokes. The contour phase occasionally involves extremely short strokes, whose entropy values are all close to zero. For these tiny strokes, sketching entropy does not provide useful ordering. We thus set w_1 to a small positive value (we use 0.05-0.1) for the contour phase, so that stroke ordering can also be guided by the Gestalt rules. By using the above cost function as weight, we adopt the greedy Prim's minimum-spanning-tree algorithm to extract the minimum cost edges until a tree is extracted from the graph. We perform this first at the level of clustered nodes to get part-level orders and then at the level of primitive nodes for stroke-level ordering.

Often, part-level orders across adjacent phases should be very similar but not exactly the same. To achieve such an effect, we add virtual edges with small values between clustered nodes representing the same object part across adjacent phases. Then the virtual edges act like normal edges when exploring for the order.

4 Experimental Results and Discussions

We have applied our technique to a variety of organic models and produced interesting dynamic sketching animations. Figures 1, 7, and 8 show snapshots of such animations. The segmented input models can be found in Figure 6. For the synthesized sketching animations please watch the accompanying video. Our results were synthesized on an Acer Aspire laptop (Dual Core @1.90GHz and 2.40GHz, 6GB RAM) . The computational bottleneck of our algorithm lies in the entropy computation step, which takes 0.060 seconds on average for a typical stroke.

User Study. We conducted a user study to evaluate the stroke ordering produced with our sketching entropy formulation (Equation 4). We compared it with the ordering generated using Gestalt rules alone [Fu et al. 2011] in terms of visual plausibility. We simulated the sketching of the five models, shown in Figure 1 and 8 except the hand model. The cloth-over-two-boxes model does not have a natural skeletal representation. We thus disabled the posture phase for this model. To facilitate the ease of examining the ordering difference, we separated each animation into two videos, one for the conceptual stage (posture and primitive phases) and the other for the perceptual stage (contour and detail phases). Note that for the conceptual stage, $w_1 \equiv 0$ in Equation 4 and thus the sketching entropy completely controls the ordering of strokes. In total, we had 5 (models) $\times$ 2 (stages) $\times$ 2 (methods) $=$ 20 videos.

We had 74 people with ages ranging from 16 to 40 to help with the user study. Among them, 55.4 % were female and 41.9% had received professional drawing training. Each participant was asked to rate the visual plausibility of each video on a discrete scale from 1 (worst) to 5 (best). Repeated measure analysis of variance (ANOVA) found a statistically significant difference in the rating score between the two ordering strategies ($p < 0.006$). The average score of our method (mean=3.44, var=0.05) was slightly higher than that of the Gestalt-driven method (mean=3.24, var=0.08).

We also checked if people with different drawing experience have different opinions on the synthesized results. The ANOVA found a significant effect ($p < 0.002$) of the drawing skill level on the rating. The participants with drawing trainings had a clearer preference for our method. On average their score for our method was 10.14% higher than that for the Gestalt-driven method. In contrast, the participants with limited drawing skills had no clear preference between the two drawing strategies. We observed that their average score for our method was only 3.04% higher. The difference coincides with Sommers' conclusion that Gestalt rules are only proved applicable for drawing simple objects, for people with no professional drawing training [van Sommers 1984].

Strokes in latter phases usually provides little or even no new information. Our sketching entropy effectively avoid the drawing of repeated lines, i.e., lines with no new information. In contrast, the Gestalt rules have no such mechanism, possibly making the drawing cursor moves around but drawing nothing new. The effectiveness of clutter removal by sketching entropy is confirmed by a significantly smaller number of the finally drawn strokes (around 30% less) and shorter animations (around 7% shorter on average).

Limitations. The effectiveness of our approach relies on the quality of input segmentation. In addition, our drawing phases are designed based on the prior knowledge and cannot be automatically customized for different users. A more flexible solution might be achieved using a data-driven approach.

5 Conclusion and Future Work

The study we have reported is the first to tackle the problem of emulating the process of drawing by observation. We introduce the concept of sketching entropy, enabling the selection and ordering of strokes, which are vital to the automatic synthesis of visually plausible dynamic sketching animations from a static 3D model with semantic segmentation. It would be interesting to extend our technique to simulate the creation process of other stroke-based media like ink painting, where the drawing order plays an important role in non-photorealistic rendering of strokes. We are also interested in applying the introduced concept of sketching entropy to other applications such as clutter removal towards line drawing simplification. Besides our current shape-based entropy definition we would also explore other types of information such as color and motion to define sketching entropy in the future.

Acknowledgements

We thank the reviewers for their constructive comments and the user study participants for their time. This work was partially supported by grants from the RGC of HKSAR (No. GRF 619012 and 16209514) and the City University of Hong Kong (No. 7002925).

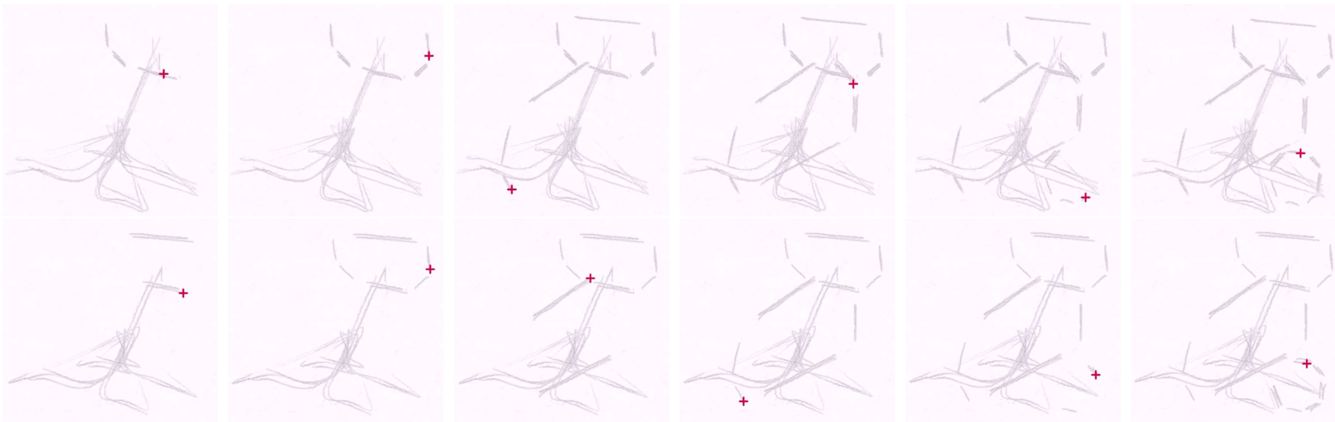


Figure 7: *Top: ordering results using Gestalt rules only. Bottom: ordering results using our sketching entropy formulation alone. Observe that the sketching entropy approach is able to implicitly group the nearly parallel strokes together without using Gestalt rules.*

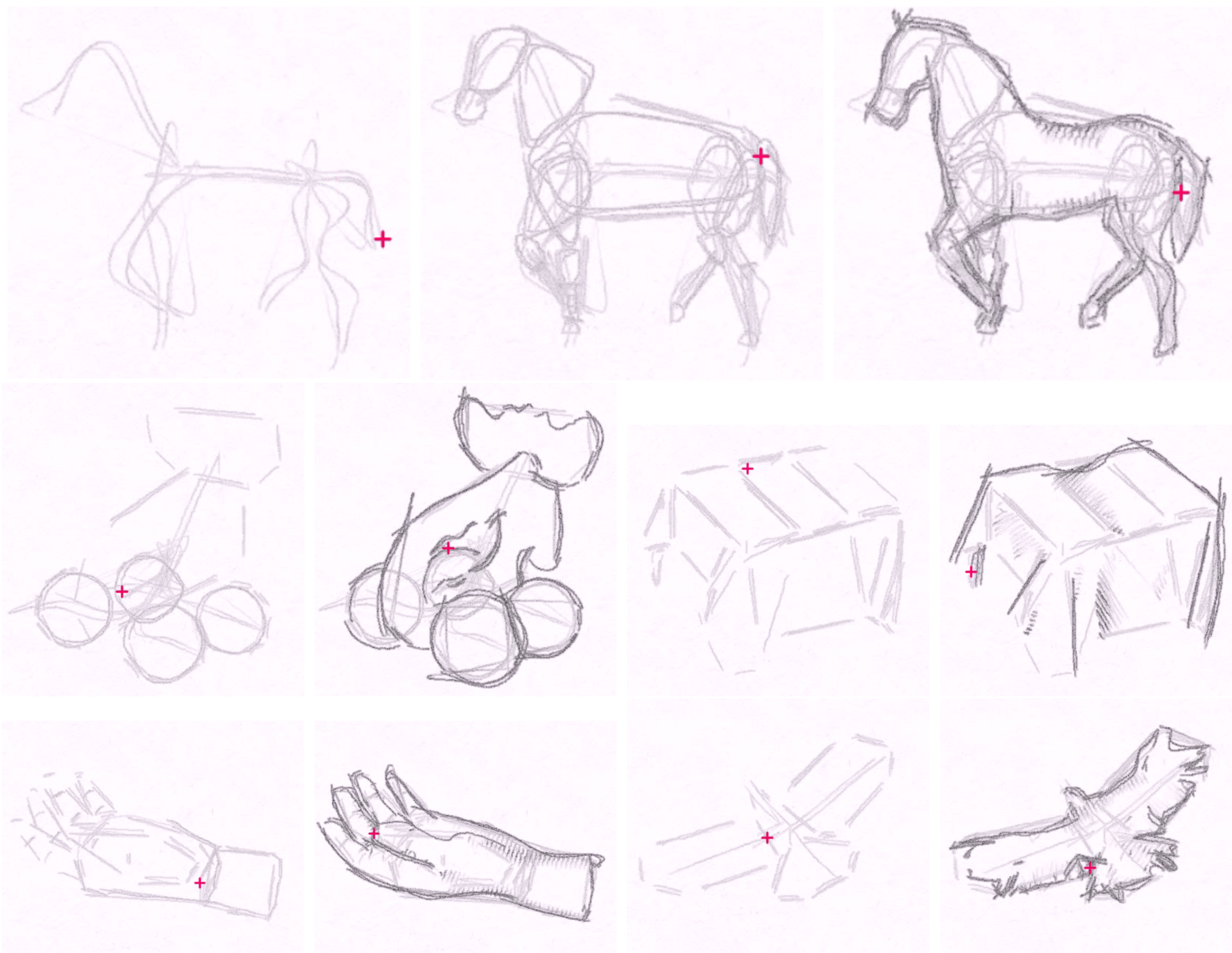


Figure 8: *Snapshots of sketching animations used in the user study. See Figure 6 for the input segmented models and the accompanying video for the dynamic sketching animations.*

References

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*, 3 (Aug.), 1.

DECARLO, D., AND RUSINKIEWICZ, S. 2007. Highlight lines for conveying shape. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering - NPAR '07*, ACM Press, New York, New York, USA, vol. 22, 63.

FU, H., ZHOU, S., LIU, L., AND MITRA, N. J. 2011. Animated construction of line drawings. *ACM Transactions on Graphics 30*, 6 (Dec.), 1.

GRABLI, S., DURAND, F., AND SILLION, F. 2004. Density measure for line-drawing simplification. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, IEEE, vol. 2, 309–315.

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 (Mar.), 1–20.

GUAY, M., CANI, M.-P., AND RONFARD, R. 2013. The line of action. *ACM Transactions on Graphics 32*, 6 (Nov.), 1–8.

HOUSE, D. H., AND SINGH, M. 2007. Line Drawing as a Dynamic Process. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)*, IEEE, 351–360.

IARUSSI, E., BOUSSEAU, A., AND TSANDILAS, T. 2013. The drawing assistant. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*, ACM Press, New York, New York, USA, 183–192.

KALOGERAKIS, E., NOWROUZEZAHRAI, D., BRESLAV, S., AND HERTZMANN, A. 2012. Learning hatching for pen-and-ink illustration of surfaces. *ACM Transactions on Graphics 31*, 1 (Jan.), 1–17.

KAVAKLI, M., SCRIVENER, S. A., AND BALL, L. J. 1998. Structure in idea sketching behaviour. *Design Studies 19*, 4 (Oct.), 485–517.

LIEN, J.-M., KEYSER, J., AND AMATO, N. M. 2006. Simultaneous shape decomposition and skeletonization. In *Proceedings of the 2006 ACM symposium on Solid and physical modeling - SPM '06*, ACM Press, New York, New York, USA, 219.

MATTESI, M. 2011. *Force: Animal Drawing: Animal Locomotion and Design Concepts for Animators*, 1 ed. Focal Press.

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

RUSINKIEWICZ, S., COLE, F., DECARLO, D., AND FINKELSTEIN, A. 2008. Line drawings from 3D models. In *ACM SIGGRAPH 2008 classes on - SIGGRAPH '08*, ACM Press, New York, New York, USA, 1.

SHANNON, C. E. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal 27*, 3 (July), 379–423.

SUWA, M. 2003. Constructive perception: Coordinating perception and conception toward acts of problem-finding in a creative experience. *Japanese Psychological Research 45*, 4, 221–234.

TEAM, W. F. C. 2005. *The Art of Basic Drawing*. Walter Foster.

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

TVERSKY, B., AGRAWALA, M., HEISER, J., LEE, P., HANRAHAN, P., PHAN, D., STOLTE, C., AND DANIEL, M.-P. 2006. Cognitive Design Principles : from Cognitive Models to Computer Models. *Model-based reasoning in science and engineering*, 1–20.

VAN SOMMERS, P. 1984. *Drawing and cognition : descriptive and experimental studies of graphic production processes*. Cambridge University Press.

ZHENG, Y., TAI, C.-L., AND AU, O. K.-C. 2012. Dot scissor: a single-click interface for mesh segmentation. *IEEE transactions on visualization and computer graphics 18*, 8 (Aug.), 1304–12.

ZHENG, Y., TAI, C.-L., ZHANG, E., AND XU, P. 2013. Pairwise harmonics for shape analysis. *IEEE transactions on visualization and computer graphics 19*, 7 (July), 1172–84.

Appendix

Enhanced PID Control System

We have made several improvements to the original linear PID control system [House and Singh 2007]. Firstly, observing that there is often a retracing effect when artists draw ovals, we concatenate several loops, to generate the retracing effect. Secondly, the retracing of an oval is often a process of refinement, from quicker and looser strokes to slower and more careful ones. To model this effect, we start the control system with a large initial value (quick) for $v(t)$ and a small value (loose) for $\omega(t)$, where $v(t)$ is the velocity of a virtual moving pen at time t and $\omega(t)$ is the natural frequency of the pen controller system. As the sketching goes on, we smoothly decrease the velocity and increase the natural frequency. Like [House and Singh 2007], we use the following formulation to compute the acceleration of the pen

$$\frac{\mathrm{d}^2 p}{\mathrm{d}t^2} = -\omega(t)^2 e - 2\zeta(t)\omega(t)\frac{\mathrm{d}e}{\mathrm{d}t} - v(t)\int e\,\mathrm{d}t, \tag{5}$$

where p is the pen's position, $\zeta(t)$ is its damping ratio, and e is the PID controller error represented as the difference between the pen and tracker positions.

Thirdly, the original PID system has an oscillation defect, caused by the uniform sampling of the track to get the stationary points. If the pen moves fast, and the values of the natural frequency and damping ratio are small, the pen will be dragged back and forth around these stationary points. We solve this problem by computing the acceleration of the pen every time, and using the closest point on tracker as the stationary point.

Distance Function F(t,u)

In our implementation, $F(t,u)$ in Equation 2 is defined as

$$F(t,u) = 1/(\oint_t \oint_u \frac{d\overrightarrow{t} \cdot d\overrightarrow{u}}{r^2 + \epsilon}), \tag{6}$$

where $d\overrightarrow{t}$ is the tangent vector along curve t, r is the Euclidean distance between $d\overrightarrow{t}$ and $d\overrightarrow{u}$, and ϵ is a tiny value to avoid division by 0. This formulation is similar to the Biot-Savart's law. Two strokes are analogized to electronic currents.

Conditional Sketching Entropy

$$p_s(v|C^s) = P[N(F(s,v)) - N(F(v,v)) \geq 1 | C^s]$$
$$= \begin{cases} 1 & \text{if } \exists s' \in C^s, F(s',v) \leq F(s,v) \\ \dfrac{p_s(v)}{\min_{s' \in C^s}\{p_{s'}(v)\}} & \text{otherwise.} \end{cases} \quad (7)$$

The decrease of a feature's information gain (due to C^s), is only determined by the tightest stroke drawn previously. It is also easy to prove that the product of all frequencies of a feature never exceeds 1. This bound ensures that the redundant strokes can be detected.