Supplementary Information for
Where to draw the line?

Heping Sheng[1], John Wilder[2], Dirk B. Walther[2*]

[1] School of Medicine, Boston University
[2] Department of Psychology, University of Toronto

*Corresponding author: Dirk B. Walther

Email: bernhardt-walther@psych.utoronto.ca

This PDF file includes:

Supplementary Methods
SI References
Figures S1 to S3
Table S1

Supplementary Methods

Experiment 1

Participants were seated at a computer station, equipped with a cathode ray tube monitor (Dell) as well as a Wacom Cintiq 13HD Interactive Pen Display graphics tablet. The dual-screen experiment was programmed using the Psychophysics Toolbox in Matlab (1), running under the Linux operating system. The CRT computer screen displayed instructions and the original 800x600 pixels photograph as reference during each trial, while the graphics tablet displayed a 1440x1080 pixels copy of the original reference image overlaid with a blank white sheet at 50% transparency for tracing.

Participants received both verbal and written instructions as well as a demonstration on how to use the graphics tablet at the beginning of the experiment. They started each contour line by pressing a button on the side of the graphics pen. They could draw smooth curves as well as set anchor points for quickly making straight edges while holding down the button. The latter function is similar to the Polygonal Lasso Tool in Adobe Photoshop. Traced contours were rendered in black on the semi-transparent overlay. There are 4 control buttons on the nondominant-hand side of the tablet: NEXT and FINISH for navigating between trials, UNDO for deleting the most recent contour, and SHOW for displaying the drawing without the underlying tracing image (Figure 1C). Participants were asked to draw clean outlines of the original image and to avoid extraneous features including repeated lines, shading or labels.

Participants performed four practice trials with stimuli of increasing complexity, intended to familiarize them with the tracing task and the software interface (Figure 1B). Then, before the experimental trials began, they were given the following instructions:

For every image, please annotate all important and salient lines, including closed loops (e.g., boundary of a monitor) and open lines (e.g., boundaries of a road). Our requirement is that, by looking only at the annotated line drawings, a human observer can recognize the scene and salient objects within the image.

During each trial, participants had up to 10 minutes to trace a photograph from the stimulus set. They could proceed at their own pace by finishing a drawing early, or taking a break before the start of the next trial. Non-artists completed 5 trials. Artists were asked to trace as many pictures as they could comfortably do in 2 hours. The 18 stimuli were presented in random order to minimize the influence of practice and fatigue. We recorded spatial coordinates and timing of the strokes for all lines in the final drawings.

Experiment 2

To identify which contour lines participants in Experiment 1 most agree on when depicting complex natural scenes, we developed an algorithm to match contour lines. Given two drawings of the same photograph, a human observer can easily identify lines in each drawing depicting the same element, even though the strokes may vary somewhat in length, number, spatial location and shape. Here we strive to replicate this ability algorithmically.

Let us assume that we have two line drawings, a reference (R) and a query (Q). For each contour in Q we wish to determine the contour in R that is the closest match. In our drawings, contours are composed of a contiguous set of straight line segments. To make the analysis of long line segments easier, we split segments longer than 10 pixels into equal-length shorter segments such that all line segments are at most 10 pixels long.

Next, we calculated the Euclidean distance of a given query line segment to each reference line segment. We use the center point of each line segment for this computation. The reference line segment with the shortest distance is considered a match as long the distance is at most 30 pixels, and the identifier of the corresponding reference contour is stored with the query contour segment. Any contour segments more than 30 pixels away from the closest reference contour are considered to be without a match in the reference.

This computation potentially leads to conflicting assignments along the length of the query contour. We resolve such conflicts with a two-stage voting process. First, we implemented a majority vote with a sliding window of 25 segments across the contour. That is, the reference contour identifier most frequently assigned to the individual line segments within the window was assigned to the contour segment at the center of the window. Remember that contour segments are at most 10 pixels long, so the sliding window in this procedure is at most 250 pixels wide. Second, another majority vote was then applied to all line segments within the query contour to determine the best matching reference contour. This process usually led to a unique assignment of the reference contour and only rarely led to a tie. When ties occurred, they were broken by determining which of the contending reference contours was closer to the query contour over its entire length. Specifically, we computed the sum of the distances of the individual query line segments to the corresponding reference line segment, weighted by the length of the query line segments. The identifier of the reference contour with the smallest sum was assigned to the query contour.

This matching procedure takes into consideration differences in stroke length, number and spatial location as well as minor variations in shape. This method also takes into account the possibility that one contour in the reference is depicted as more than one contour in a query, but does not account for the reverse.

We used the contour matching algorithm in order to construct a super-reference that contained the superset of all contours drawn by any of the individuals in Experiment 1 for a given photograph. In the first step, we used the highest-ranked drawing as the reference and the second highest drawing as the query and proceeded with the matching algorithm as described above. Any query contours that could not be matched to the reference were added to the reference. Next, the third-highest ranked drawing was used as the query. Any query contours that did not match any of the contours in the updated reference were added to the reference, and so on, until all contours in all drawings were accounted for. As a result, this super-reference captures the superset of all contours, i.e., all non-repeating contours for each scene (Figure 2B).

In a second round of matching, each drawing was matched to the super-reference. In this round, we counted the number of times each contour in the super-reference was matched with a contour in a query drawing. Dividing the count by the number of drawings for this particular image resulted in a normalized measure of the frequency with which a particular contour was depicted (Figure S1). We call this number the consistency score.

Experiment 3

The experiment was performed on a personal computer with Matlab and the Psychophysics Toolbox (1) running on Windows 10. A cathode ray tube monitor (Dell) was used to display the experiment at a resolution of 800x600 pixels with a refresh rate of 150 Hz. Participants were seated approximately 57 cm from the screen.

Participants performed the experiment as an add-on to another, similarly structured main experiment, which will be reported elsewhere. For the main experiment, participants were asked to categorize briefly presented line drawings of real-world scenes into six categories (beaches, city streets, forests, highways, mountains, offices) by pressing one of six buttons on a keyboard (s, d, f for the left hand and j, k, l for the right hand). The assignment of keys to categories was randomized for each participant.

Each trial of the experiment started with a 500 ms fixation period, followed by a brief presentation of the target image, followed by a perceptual mask for 500 ms. The mask consisted of randomly distributed black lines on a white background. After the mask, a blank screen was displayed until a response key was detected (Figure 2D).

In an initial training phase, images were presented for 233 ms (stimulus onset asynchrony, SOA). Participants heard a low tone when they made an error. When participants responded correctly in 17 of the last 18 trials or in 72 trials in total, they moved on to the ramping phase. In the ramping phase, the SOA was linearly decreased from 200 ms to 33 ms over the course of 54 trials. During the subsequent test phase (360 trials), the SOA was fixed to 53 ms, and participants no longer received feedback.

Following the testing phase for the main experiment, participants performed an additional 36 test trials with the half-drawings created for this experiment. Key assignment, SOA, and temporal structure of the experiment remained the same as in the main experiment (see Figure
9). Accuracy of categorizing the drawings was recorded separately for high- and low-consistency drawings and compared using a paired t test.

All data, statistical analyses and images of the drawings are available at: https://osf.io/x9uj5/

SI References

1. D. H. Brainard, The Psychophysics Toolbox. *Spatial Vision* **10**, 433–436 (1997).

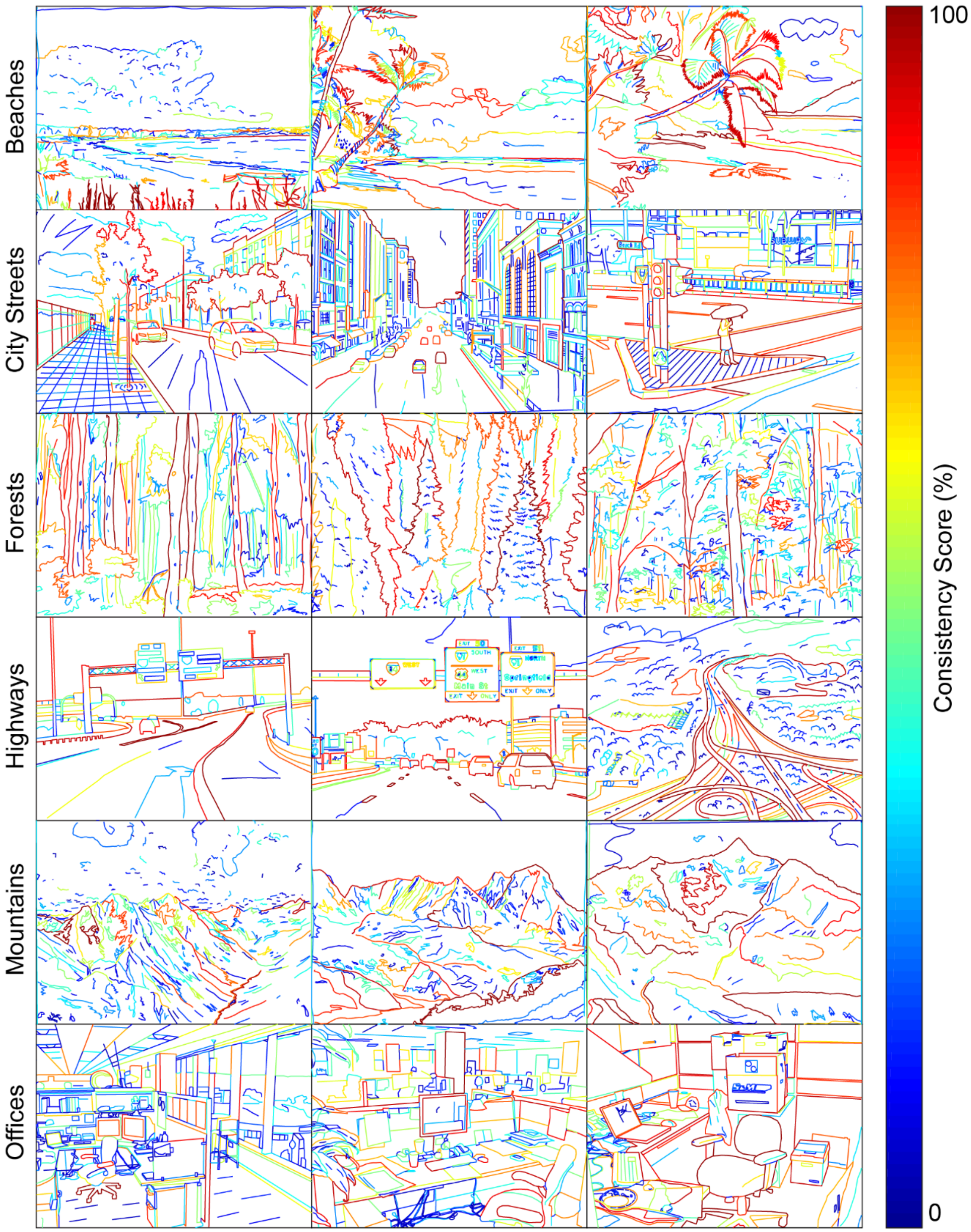

Figure S1. Super-references for all 18 scene images.

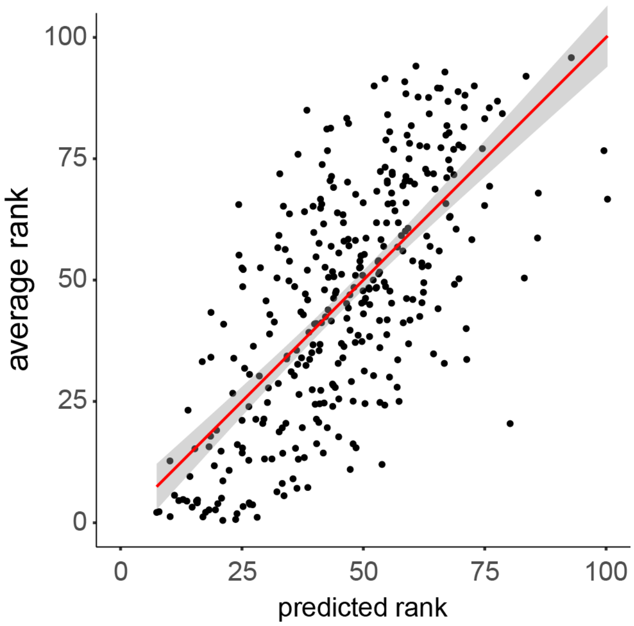

Figure S2. Average rank of drawings versus the ranks predicted by the multiple linear regression model. The regression line is shown in red with the 95% confidence interval in gray.

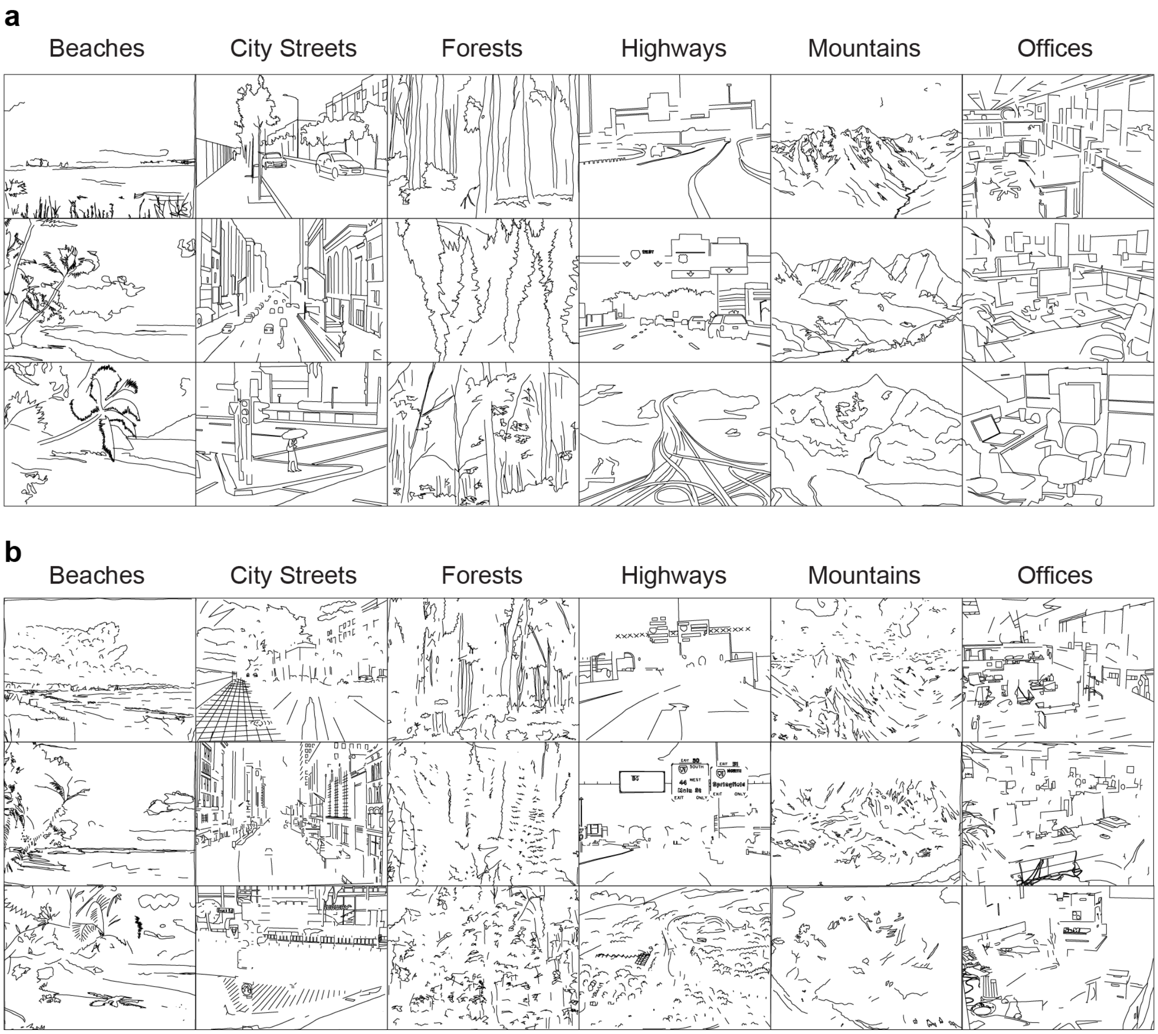

Figure S3. Stimuli for Experiment 3. (a) Most consistent half-drawings. (b) Least consistent half-drawings.

Table S1. Inter-rater agreement of line drawing rankings

Image ID	Kendall's W	df	p-value
Beach 1	0.57	18	$7.62 \cdot 10^{-12}$
Beach 2	0.67	20	$6.52 \cdot 10^{-14}$
Beach 3	0.60	17	$1.60 \cdot 10^{-10}$
City 1	0.82	21	0
City 2	0.88	22	0
City 3	0.85	21	0
Forest 1	0.79	14	0
Forest 2	0.72	20	0
Forest 3	0.64	23	0
Highway 1	0.72	20	0
Highway 2	0.68	16	$4.96 \cdot 10^{-6}$
Highway 2	0.69	22	0
Mountains 1	0.92	22	0
Mountains 2	0.87	21	0
Mountains 3	0.67	17	$2.22 \cdot 10^{-16}$
Office 1	0.85	20	0
Office 2	0.83	19	0
Office 3	0.81	20	0