Supplemental Material for papers_363 "Generating Freehand Concept Sketches from CAD Sequences"

ANONYMOUS AUTHOR(S)

1 OVERVIEW

In this document, we provide additional details about parameter selection, about the formulation of the binary integer program, about the data generation for the normal prediction, and about the evaluation with design teachers. We also provide additional results of normal prediction on ABC models, OpenSketch models and our synthetic models.

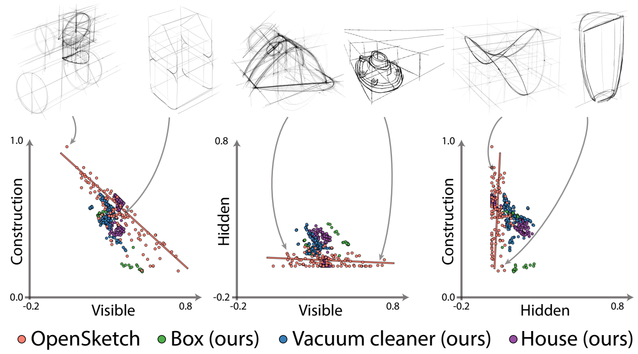

Fig. 1. Real sketches from OpenSketch (top and orange dots) lie along a linear sub-space when considering their proportion of construction lines, visible feature lines, and hidden feature lines. In contrast, the sketches generated by our method with varying λ_1 and λ_2 parameters can be far from this sub-space (green, blue and purple dots, which correspond to three different CAD models). For a given CAD model, we select the values of λ_1 and λ_2 that produce sketches close to the 3D line regressed from real sketches (orange line).

2 PARAMETER SELECTION

Our method allows users to control construction quality and visual clutter by varying λ_1 and λ_2.

To further ground our algorithm in real-world practice, we studied how the results obtained with different parameter settings compare to real sketches. In the absence of ground-truth 3D lines for OpenSketch data, we cannot evaluate our score function on real examples. However, we can leverage OpenSketch annotations to compute statistics about different types of lines, and see if these statistics match the ones we obtain in our synthetic sketches.

For each of the first-view concept sketches in OpenSketch, we computed the number of strokes labeled as constructions lines (C), the number of strokes labeled as visible feature lines (V), and the number of strokes labeled as hidden feature lines (H). We then associate each sketch with a feature vector $\left(\frac{C}{N}, \frac{V}{N}, \frac{H}{N}\right)$, where N is the total number of lines in that sketch. Figure 1 (orange) plots the distribution of sketches in that feature space, which reveals that real-world sketches cluster along a line, such that the proportion of visible lines varies linearly with the proportion of construction lines, while the proportion of hidden lines is low overall. In contrast, Figure 1 (blue, green and purple) shows that the sketches generated by our method for varying λ_1 and λ_2 can lie far away from the linear distribution of real sketches, in particular by exhibiting a greater proportion of hidden lines.

Based on this observation, we propose a simple automatic parameter tuning procedure. For a given input CAD model, we run our algorithm multiple times with λ_1 and λ_2 parameters regularly distributed over their respective interval. We then favor the results for which the feature vector is close to the 3D line fitted onto the cluster of real sketches, as measured by the distance D_{real}. However, many of these results exhibit the same ratios of construction and feature lines but contain a different quantity of lines. Among those, we favor the results that contain the greatest number of visible feature lines, as measured by $D_{\text{visible}} = 1 - \frac{V_{\lambda_1,\lambda_2}}{V_{\text{total}}}$ where V_{λ_1,λ_2} is the number of visible feature lines selected for a given choice of (λ_1, λ_2) parameters, and V_{total} is the total number of visible feature lines generated initially. Combining these two criteria, we select the result that minimizes $D = D_{\text{real}} + D_{\text{visible}}$.

3 INTEGER PROGRAM FORMULATION FOR LINE SELECTION

In the following, we give additional details for the integer program formulation from the main paper. Its full implementation in Gurobi will be released with the paper.

Anchoring. The anchoring term for curves also favors the presence of *tangential* 3D intersections with other curves or straight lines. Denoting $\{i_p^k\}$ the set of tangential intersections along s_p and $\mathcal{S}_p^{k^{tangent}}$ the set of tangential lines going through i_p^k, we condition the selection of $\mathbf{f}_p^k$ by the selection of tangential lines:

$$\sum_{q \in \mathcal{S}_p^{k^{tangent}}} \mathbf{s}_q \geq \mathbf{f}_p^k \tag{1}$$

Projections. If s_p is a projection line, we extract the subgraph $\mathcal{G}_p^Q$ from the intersection graph, including all strokes lying in the major-axis plane Q that is containing s_p. In this planar subgraph, the goal is to build a formulation which checks the existence of a path starting from the point s_p^{start} and ending in s_p^{end}. The existence of such a path is validated by the selection of $\mathbf{p}_p$

We define a source variable $\mathbf{v}^{source}$ and a sink variable $\mathbf{v}^{sink}$, representing s_p^{start} and s_p^{end}, respectively. For the rest of the nodes $v_i \in \mathcal{G}_p^Q$, we define the binary variables $\mathbf{v}_i$.

Intuitively, the idea of the following constraint system is the following. To find a path, we want to select a subset of nodes v_i. Every node included in the path should have exactly one selected *incoming* neighbour and one selected *outgoing* neighbour. Except for the source and the sink, which should have exactly one outgoing neighbour and one incoming neighbour, respectively.

For the source, we check if exactly one outgoing node has been chosen.

$$\sum_{i \in v^{source}_{outgoing}} \mathbf{v}_i \leq 1 + M(1 - \mathbf{p}_p) \tag{2}$$

$$\sum_{i \in v^{source}_{outgoing}} \mathbf{v}_i \geq 1 - M(1 - \mathbf{p}_p) \tag{3}$$

Similarly, for the sink node, exactly one incoming node has to be chosen.

$$\sum_{i \in v^{sink}_{incoming}} \mathbf{v}_i \leq 1 + M(1 - \mathbf{p}_p) \tag{4}$$

$$\sum_{i \in v^{sink}_{incoming}} \mathbf{v}_i \geq 1 - M(1 - \mathbf{p}_p) \tag{5}$$

And for every other node, the number of incoming and outgoing neighbours should be exactly the same.

$$\sum_{i \in v^{q}_{incoming}} \mathbf{v}_i - \sum_{i \in v^{q}_{outgoing}} \mathbf{v}_i \geq 0 - M(1 - \mathbf{v}_q), \forall v_q \in \mathcal{G}^Q_p \tag{6}$$

$$\sum_{i \in v^{q}_{incoming}} \mathbf{v}_i - \sum_{i \in v^{q}_{outgoing}} \mathbf{v}_i \leq 0 + M(1 - \mathbf{v}_q), \forall v_q \in \mathcal{G}^Q_p \tag{7}$$

Intuitively, the condition that only one neighbouring node should be selected will be propagated from the source and sink nodes and every node in between will select either zero neighbouring nodes or one incoming and one outgoing nodes.

Additionally, all nodes should be fully anchored.

$$\mathbf{v}_q \leq \mathbf{f}_q, \forall v_q \in \mathcal{G}^Q_p \tag{8}$$

And finally, the corresponding stroke variables for all nodes precede the node selection.

$$\mathbf{v}_q \leq \mathbf{s}_q, \forall v_q \in \mathcal{G}^Q_p \tag{9}$$

4 NORMAL MAP PREDICTION

Shape grammar. We generated our training dataset by synthesizing CAD sequences according to a simple shape grammar. Each sequence is compose of four steps:

$$\begin{aligned}
&\text{Extrude}_{\square} \\
&(\text{Extrude}_{\{(+|-),(\square|\circ)\}}|\text{Chamfer}) \\
&(\text{Extrude}_{\{(+|-),(\square|\circ)\}}|\text{Chamfer}) \\
&\text{Fillet}
\end{aligned}$$

The sequence always starts by a square extrusion to create a cuboid. We then perform either a positive extrusion to create a protrusion, a negative extrusion throughout the shape to create a hole, or a chamfer to create slanted surfaces. Square or disk profiles are used for these operations. We repeat this step once, and end with a fillet operation on a variable number of edges. Extrusion faces, chamfer edges and fillet edges are randomly chosen from the intermediate BREP. Extrusion depth and fillet radius are randomly picked from a set of discrete values. The size and position of the square or disk profiles are also randomly sampled.

Viewpoints. To render our synthetic CAD sequences, we selected viewpoints that best show the involved CAD operations, where we used the dot product between the view direction and the operation direction to test if the operation is well seen.

Additional Results. We provide additional normal prediction results on ABC models (108), OpenSketch models (12 drawings from designer $D3$ and 12 drawings from designer $D1$), and a small random subset of our synthetic models (100) in the supplemental folder, where there is a readme file demonstrating how to read the results.

5 EVALUATION BY DESIGN TEACHERS

We provide the detailed teacher evaluation forms in the supplemental folder. Figure 2 visualizes individual ratings for each criteria and each sketch.

The bottom part of the figure compares the three variants of our method (*M1*, *M2*, and *Ours*). While *M1* is consistently judged inferior, *M2* is close to *Ours*. *R3* has a preference for our results over the ones produced by *M2* (see *construction* for the house, *amount of details* for the box, and *line execution* for the vaccum cleaner and for the box). In contrast, *R1* judged *M2* better for the house but inferior for the box. *R2* gave the same score to both variants in most cases. Note that similar disagreement happens for real sketches, such as about the construction of the house for D_2 and of the box for D_2 and D_3.

Finally, note that *R1* gave lower scores overall (mean score of 2.11 against 2.58 for *R2* and 2.69 for *R3*, computed over all 18 sketches).

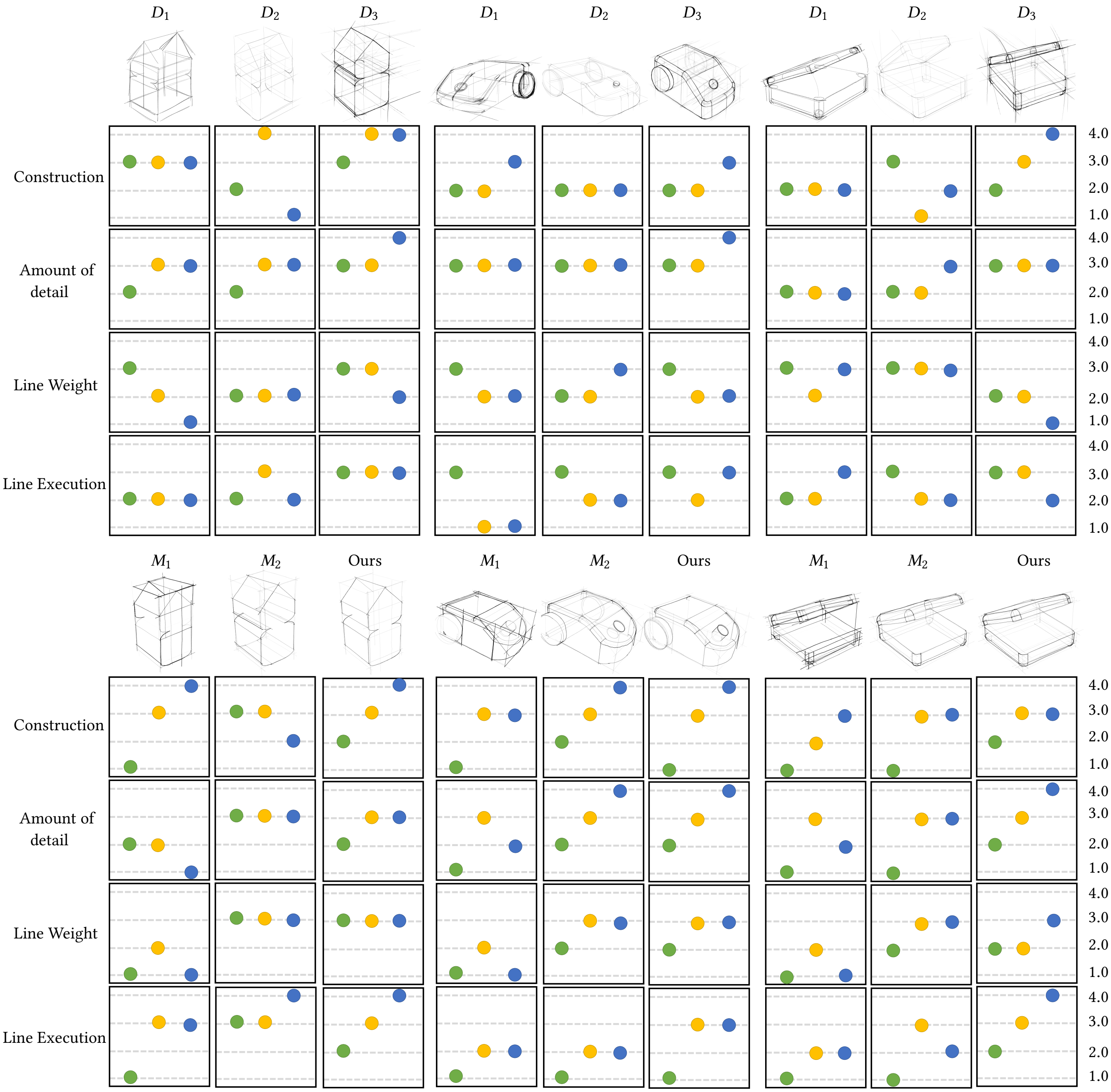

Fig. 2. **Teacher Evaluation Results.**