Polygon Detection for Room Layout Estimation using Heterogenous Graphs and Wireframes - Supplementary Material

David Gillsjö Gabrielle Flood Kalle Åström
Centre for Mathematical Sciences, Lund University, Sweden
{david.gillsjo,gabrielle.flood,kalle.astrom}@math.lth.se

S1. Introduction

This is supplementary material for the paper "Polygon Detection for Room Layout Estimation using Heterogenous Graphs and Wireframes". It includes a more thorough explanation of the cycle generating algorithm, the maximum average weight cycle algorithm and statistics for inference time.

S2. Cycle Generation

The CSP model described in section 3.3 generated polygons from the wireframe detections which are later classified. This section describes how the polygons are generated. Take the wireframe junctions and lines and form a homogeneous undirected graph $\mathcal{G}$ where each vertex is a junction and each edge is a line. In short we first find all cycles, then we check for uniqueness, and finally we make sure each cycle is a polygon without self-intersections. From $\mathcal{G}$ we find all connected subgraphs $\mathcal{S} = \{\mathcal{G}_s | \mathcal{G}_s \in \mathcal{G}\}$, which are disjoint. For each subgraph $\mathcal{G}_s$ we find the cycle basis $\mathbf{C}_s = \{C_1, C_2, ...C_Q\}$, consisting of cycles. With the cycle basis it is possible to generate a cycle [3] in $\mathcal{G}_s$ by taking any connected subset $\Theta_k \subseteq \mathbf{C}_s$ such that each $C_i \in \Theta_k$ is connected to at least one of the other cycles, i.e

$$C_i \cap (\Theta_k \setminus C_i) \neq \emptyset. \tag{1}$$

The cycle is formed by taking the XOR (exclusive disjunction) product of all base cycles in $C_i \in \Theta_k$. Let

$$Z = X \oplus Y \tag{2}$$

denote the XOR product between the two graphs X and Y. Then the vertices V_z of Z will be the cartesian product of the vertices V_X, V_Y from X and Y respectively, i.e. $V_Z = V_X \times V_Y$. For all vertex pairs $(v_i, v_j) \in V_Z$ we form an edge $e = (v_i, v_j)$ in Z if $e \in X$ or $e \in Y$ but not if $e \in X \cap Y$. From the chosen basis subset $\Theta_k = \{C_1, C_2, ..., C_R\}$ we form a new cycle

$$C_k = C_1 \oplus C_2 \oplus ... \oplus C_R \tag{3}$$

and check if the generated cycle C_k has at least 3 vertices and is a geometric valid polygon without intersections. If so we say that C_s is a valid polygon. To generate all polygons this is iterated for all permutations of connected subsets in each connected subgraph $\mathbf{C}_s$.

Calculating the cycle bases is cheap compared to generating all cycles, so this approach saves time during training since we only generate a fixed amount of polygons during training. For inference however we must find all polygons, which does not scale well.

S3. Maximum Average Weight Cycle

For the HGC model described in section 3.4 we need to find the best cycle from the neural networks edge scoring in the proposal step. We want to find the cycle C^* in a homogeneous undirected graph $\mathcal{G}$ with the maximum average edge weight. While there are many methods for finding the minimum average edge weight cycle in a directed graph, for example the algorithm by Karp [2], we did not manage to restrict it to cycles with at least three edges. Which is a problem since the optimal cycle will be traversing the highest scoring edge twice and be done. Therefore we use a greedy method based on the shortest path algorithm of Djikstra [1].

For each plane anchor A_k we form a graph $\mathcal{G}_k$ were each edge $e_j \in \mathcal{G}_k$ correspond to detected line L_j and each vertex $v_i \in \mathcal{G}_k$ correspond to an junction J_i. For an edge e_j we calculate a score

$$s_j = \sigma\left(\hat{\mathbf{g}}_j^T W \hat{\mathbf{d}}_k\right), \tag{4}$$

as explained in section 3.4.2. From the score each edge is given a weight $w_i = 1 - s_i$ to formulate the problem as finding the minimum average weight cycle.

The algorithm is outlined in Algorithm S1 and will iteratively try a different edge $e_s \in \mathcal{G}_k$ as starting point for minimum weight cycles. The cycle is found by taking the vertices of the edge as start respectively target of the shortest path problem on the graph $\mathcal{G}_k \setminus e_s$. By finding the shortest path and adding e_s we have a minimum weight cycle containing the starting edge. Edges are tried as starting edge going from lowest to highest weight for a fixed amount of iterations T.

S1 Algorithm: An overview of how the algorithm finds an approximate minimum average weight polygon.

Data: Graph : $\mathcal{G}_k$,
Edges and weights: $(e_j, w_j) \in \mathcal{G}_k$,
Iterations: T
Result: Polygon C^* with approximate lowest average weight w^*
$E := \{e_j \mid (e_j, w_j) \in \mathcal{G}_k\}$, candidate edges
$w^* := \infty$
$C^* := \emptyset$
for $t = 1, ..., T$ **do**
  $e_t :=$ `takeMinWeightEdge(`E`)`
  $E := E \setminus e_s$
  $\mathcal{G}_t := \mathcal{G}_k \setminus e_t$
  $C_t :=$ `shortestPath(`$\mathcal{G}_t, e_t[1], e_t[2]$`)`
  $w_t := \text{mean}(\{w_j \mid (e_j, w_j) \in C_t\})$
  if $w_t < w^*$ **and** `numberOfEdges(`C_t`)` ≥ 3 **and** `isValidPolygon(`C_t`)` **then**
    $w^* := w_t$
    $C^* := C_t$
  end
end

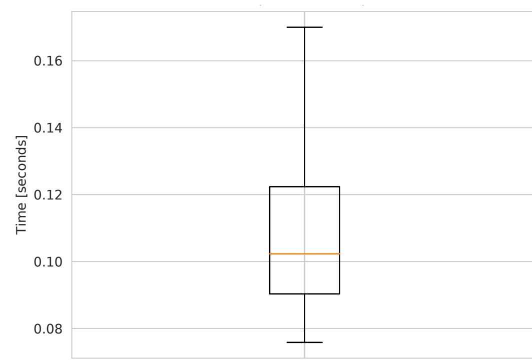
S4. Inference Time

We measure inference time by evaluating the model over 1000 images in the validation set on an NVidia Titan V GPU. Measuring from having the image in RAM to getting the result back on the GPU. In table S1 we see that the Graph-based HGC method is much faster on average while also having much smaller variations in inference time. See boxplots in Figure S1 for a visualization.

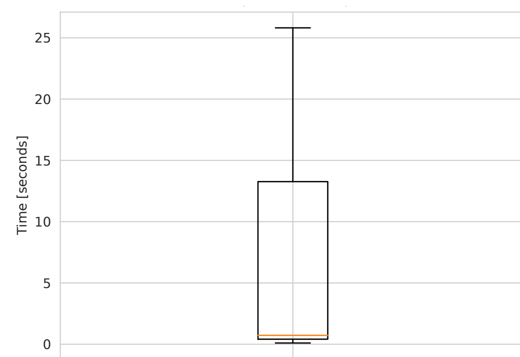
Table S1: Median and mean inference times in seconds with standard deviation for the CSP and HGC model.

Model	Median	Mean	Std
CSP	8.88	0.72	11.98
HGC	0.10	0.11	0.06
HGC +	0.12	0.12	0.06

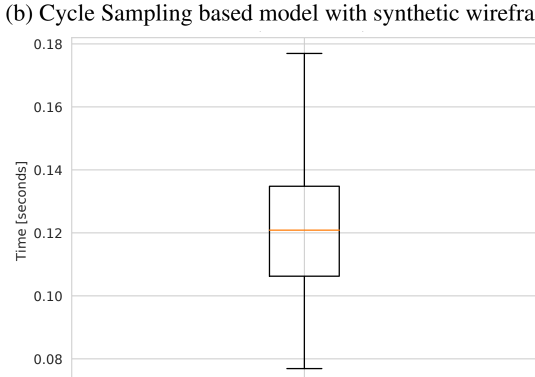
+ Joint Wireframe Detection.



(a) Heterogeneous Graph model with synthetic wireframe.



(b) Cycle Sampling based model with synthetic wireframe.



(c) Heterogeneous Graph model with predicted wireframe.

Figure S1: Inference times for the two different models. HCG is tested with synthetic and joint wireframe prediction.

References

[1] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959. 1

[2] Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23(3):309–311, 1978. 1

[3] Boon Chai Lee. Algorithmic approaches to circuit enumeration problems and applications. Technical report, Cambridge, Mass.: Massachusetts Institute of Technology, Dept. of ..., 1982. 1