Supplementary Document for “LARGE: A Length-Aggregation-based Grid Structure for Line Density Visualization”

Tsz Nam Chan[1], Bojian Zhu[2], Dingming Wu[1], Yun Peng[3], Leong Hou U[4]

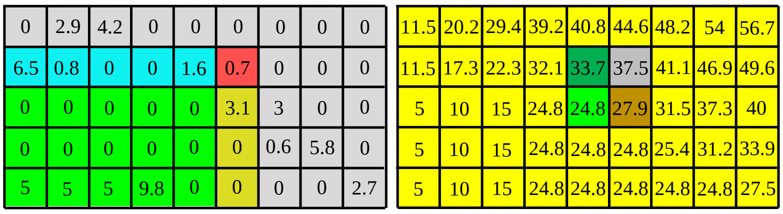
[1]Shenzhen University, [2]Hong Kong Baptist University,
[3]Guangzhou University, [4]University of Macau

In this supplementary document, we further include some content that cannot be presented in the paper [3] due to space limitations.

I Prefix-Sum Grid Structure (PG): Construction and Computation of Aggregate Values

In order to efficiently construct the prefix-sum grid structure (i.e., efficiently obtain $PG[\alpha^*, \beta^*]$ in Equation 8 of [3]), we need to adopt Equation I to compute $PG[\alpha^*, \beta^*]$ for each (α^*, β^*)-pair, by following the lexicographical order of (α^*, β^*) (from small to large).

$$PG[\alpha^*, \beta^*] = \begin{cases} G[\alpha^*, \beta^*] & \text{if } \alpha^* = 1, \beta^* = 1 \\ PG[\alpha^* - 1, \beta^*] + G[\alpha^*, \beta^*] & \text{if } \alpha^* \neq 1, \beta^* = 1 \\ PG[\alpha^*, \beta^* - 1] + G[\alpha^*, \beta^*] & \text{if } \alpha^* = 1, \beta^* \neq 1 \\ PG[\alpha^* - 1, \beta^*] + PG[\alpha^*, \beta^* - 1] & \\ -PG[\alpha^* - 1, \beta^* - 1] + G[\alpha^*, \beta^*] & \text{if } \alpha^* \neq 1, \beta^* \neq 1 \end{cases} \quad \text{(I)}$$



0	2.9	4.2	0	0	0	0	0	0
6.5	0.8	0	0	1.6	0.7	0	0	0
0	0	0	0	0	3.1	3	0	0
0	0	0	0	0	0	0.6	5.8	0
5	5	5	9.8	0	0	0	0	2.7

(a) Grid structure with the accumulated length for each grid

11.5	20.2	29.4	39.2	40.8	44.6	48.2	54	56.7
11.5	17.3	22.3	32.1	33.7	37.5	41.1	46.9	49.6
5	10	15	24.8	24.8	27.9	31.5	37.3	40
5	10	15	24.8	24.8	24.8	25.4	31.2	33.9
5	10	15	24.8	24.8	24.8	24.8	24.8	27.5

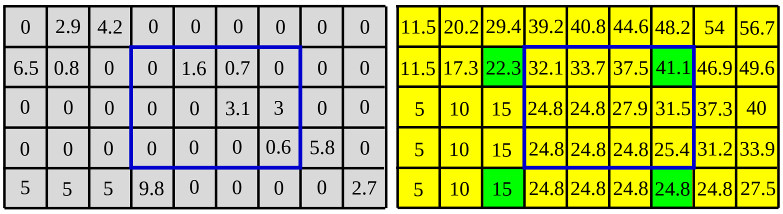
(b) Prefix-sum grid structure (based on length-aggregation)

Figure I: Illustration of constructing the prefix sum grid structure PG in (b) for the grid structure G in (a).

Using Figure I as an example, the grey entry 37.5 of PG is the same as 33.7 (the dark green entry of PG, which represents the sum of all values in the blue and light green regions of G) + 27.9 (the brown entry of PG, which represents the sum of all values in the light green and orange regions of G) - 24.8 (the light green entry of PG, which represents the sum of all values in the light green region of G) + 0.7 (the red entry of G). Since each entry of PG can be computed in $O(1)$ time (by accessing at most four entries), the time complexity of constructing the prefix-sum grid structure PG is $O(XY)$ (cf. Lemma 3 in [3]).

With this prefix-sum grid structure PG, we can compute the aggregation of all lengths in any rectangular region of grids, $\alpha_L \leq \alpha \leq \alpha_U$ and $\beta_L \leq \beta \leq \beta_U$, (e.g., the blue rectangle in Figure IIa) by accessing at most four grids (e.g., the green grids in Figure IIb) in the prefix-sum grid structure (cf. Equation II).

$$\sum_{\alpha=\alpha_L}^{\alpha_U} \sum_{\beta=\beta_L}^{\beta_U} G[\alpha,\beta] = \begin{cases} PG[\alpha_U,\beta_U] & \text{if } \alpha_L = 1, \beta_L = 1 \\ PG[\alpha_U,\beta_U] - PG[\alpha_L - 1,\beta_U] & \text{if } \alpha_L \neq 1, \beta_L = 1 \\ PG[\alpha_U,\beta_U] - PG[\alpha_U,\beta_L - 1] & \text{if } \alpha_L = 1, \beta_L \neq 1 \\ PG[\alpha_U,\beta_U] + PG[\alpha_L - 1,\beta_L - 1] & \\ -PG[\alpha_L - 1,\beta_U] - PG[\alpha_U,\beta_L - 1] & \text{if } \alpha_L \neq 1, \beta_L \neq 1 \end{cases} \quad \text{(II)}$$



0	2.9	4.2	0	0	0	0	0	0
6.5	0.8	0	0	1.6	0.7	0	0	0
0	0	0	0	0	3.1	3	0	0
0	0	0	0	0	0	0.6	5.8	0
5	5	5	9.8	0	0	0	0	2.7

(a) Grid structure with the accumulated length for each grid

11.5	20.2	29.4	39.2	40.8	44.6	48.2	54	56.7
11.5	17.3	22.3	32.1	33.7	37.5	41.1	46.9	49.6
5	10	15	24.8	24.8	27.9	31.5	37.3	40
5	10	15	24.8	24.8	24.8	25.4	31.2	33.9
5	10	15	24.8	24.8	24.8	24.8	24.8	27.5

(b) Prefix-sum grid structure (based on length-aggregation)

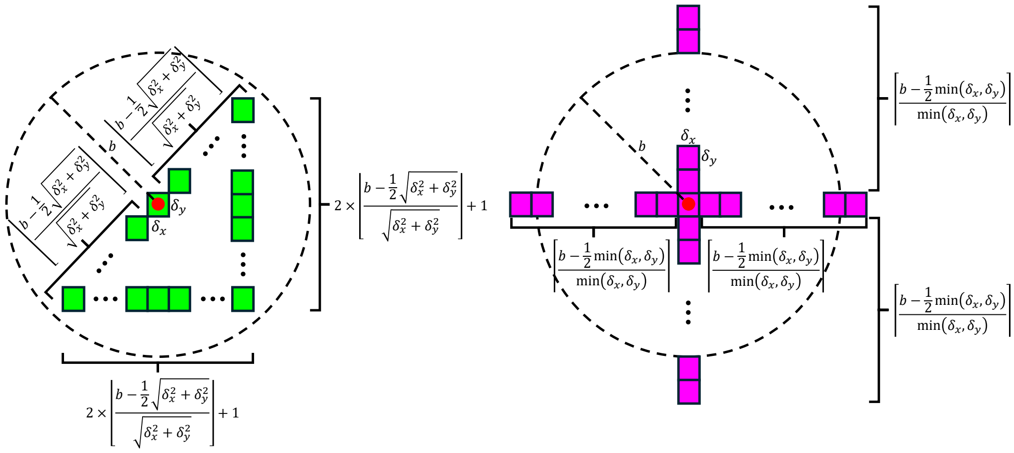
Figure II: Illustration of computing the aggregate values with any rectangular region in the grid structure G using the prefix-sum grid structure PG.

Therefore, the time complexity of obtaining the aggregation of all lengths in any rectangular region of grids (e.g., the blue rectangle in Figure IIa) in the grid structure G is $O(1)$ time (cf. Lemma 3 in [3]).

II Illustration of $A_{LB_\square(\mathbf{q})}$ and $A_{UB_\square(\mathbf{q})}$ in Section 3.6

To compute the occupied area of $LB_\square(\mathbf{q})$ (i.e., $A_{LB_\square(\mathbf{q})}$), we need to sum the areas of all those green pixels with size $\delta_x \times \delta_y$ that are completely covered by the circular search space with the bandwidth value b (cf. Figure IIIa). Based on the simple geometrical interpretation in Figure IIIa, we conclude that the total numbers of those green pixels in the horizontal direction and in the vertical direction are $2 \times \left\lfloor \frac{b - \frac{1}{2}\sqrt{\delta_x^2 + \delta_y^2}}{\sqrt{\delta_x^2 + \delta_y^2}} \right\rfloor + 1$. Therefore, we conclude that $A_{LB_\square(\mathbf{q})} = \left(2 \times \left\lfloor \frac{b - \frac{1}{2}\sqrt{\delta_x^2 + \delta_y^2}}{\sqrt{\delta_x^2 + \delta_y^2}} \right\rfloor + 1\right)^2 \delta_x \delta_y$ (which is the same as Equation 9 in [3]).

With the similar concept, we need to sum the areas of all those pink pixels that just completely cover the circular search space with the bandwidth value b (cf. Figure IIIb). Based on the simple geometrical interpretation in Figure IIIb, the smallest numbers of those pink pixels in the horizontal direction and in the vertical direction are $2 \times \left\lceil \frac{b - \frac{1}{2}\min(\delta_x, \delta_y)}{\min(\delta_x, \delta_y)} \right\rceil + 1$. Therefore, we conclude that $A_{UB_{\square}(\mathbf{q})} = \left(2 \times \left\lceil \frac{b - \frac{1}{2}\min(\delta_x, \delta_y)}{\min(\delta_x, \delta_y)} \right\rceil + 1\right)^2 \delta_x \delta_y$ (which is the same as Equation 10 in [3]).



b
$b - \frac{1}{2}\sqrt{\delta_x^2 + \delta_y^2}$
$\sqrt{\delta_x^2 + \delta_y^2}$
δ_x
δ_y
$2 \times \left\lceil \frac{b - \frac{1}{2}\sqrt{\delta_x^2 + \delta_y^2}}{\sqrt{\delta_x^2 + \delta_y^2}} \right\rceil + 1$
$\left\lceil \frac{b - \frac{1}{2}\min(\delta_x, \delta_y)}{\min(\delta_x, \delta_y)} \right\rceil$

(a) $A_{LB_{\square}(\mathbf{q})}$ (b) $A_{UB_{\square}(\mathbf{q})}$

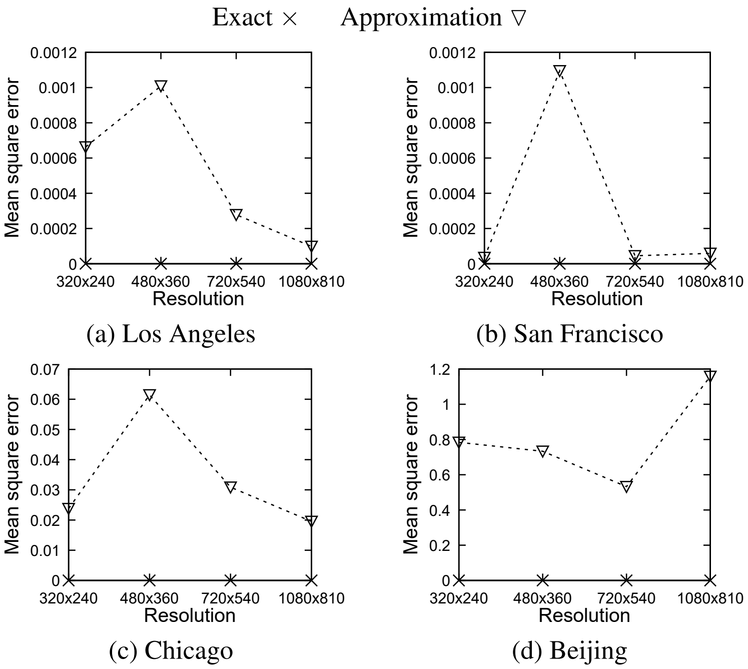
Figure III: Illustration of the occupied areas of the square-shaped bound functions (i.e., $LB_{\square}(\mathbf{q})$ and $UB_{\square}(\mathbf{q})$), where each value beside the brace denotes the number of pixels (with size $\delta_x \times \delta_y$).

III Additional Experiments

In this section, we further include two additional experiments, which are (1) accuracy of all methods for generating LDV and (2) visualization results of different approaches for generating LDV.

Accuracy of all methods for generating LDV. Here, we investigate how the resolution size affects the accuracy of exact and approximation methods. To conduct this experiment, we first set the bandwidth value and the relative error to be 1000m and 0.1, respectively, and then measure the mean square error (MSE) of each method by choosing four resolution sizes, which are 320×240, 480×360, 720×540, and 1080×810. Figure IV shows the results of all methods. Observe that the mean square error of the approximation method (i.e., LARGE) does not depend on the resolution size. The main reason is that this metric represents the line density value deviation of each pixel, which is already normalized by the number of pixels. Note that our approximation method, LARGE, still achieves small mean square error (which ranges from 9.84×10^{-5} to 1.1583) compared with the existing exact method (with zero error), no matter which resolution size we choose.

Visualization results of different approaches for generating LDV. Although the existing software platforms, including QGIS [2] and ArcGIS [1], aim to adopt Equation 4 in [3] (based on the circular search space in Figure 2 of [3]) to obtain LDV, we still do not know whether some simple approaches (e.g., simply use the bound functions in Section 3.3 and Section 3.4 of [3]) can already achieve the similar visualization results compared with the exact solution. Therefore, in this experiment, we first adopt the de-



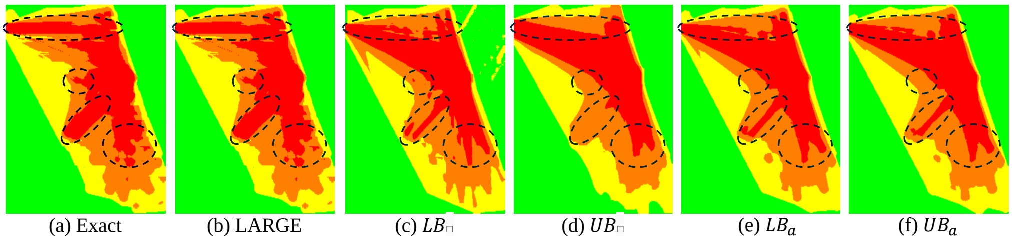
(a) Los Angeles (b) San Francisco

(c) Chicago (d) Beijing

Figure IV: Accuracy (mean squared error) for generating LDV, varying the resolution size.

fault settings (by setting the resolution size, the bandwidth parameter, and the relative error (for LARGE) to be 320 × 240, 1000m, and 0.1, respectively), and then test the visualization results with different approaches, including Exact, LARGE, LB$_\square$, UB$_\square$, LB$_a$, and UB$_a$, using the Chicago taxi mobility dataset (cf. Table 1 of [3]).

Figure V shows the visualization results of all approaches. Although LB$_\square$, UB$_\square$, LB$_a$, and UB$_a$ can theoretically provide much faster performance (cf. Theorem 3 and Theorem 4 of [3]) compared with LARGE, these approaches cannot achieve similar visualizations (cf. the black dashed curves in Figure V) compared with the exact solution (Exact) and LARGE.



(a) Exact (b) LARGE (c) $LB_\square$ (d) $UB_\square$ (e) LB_a (f) UB_a

Figure V: Generating LDV in the Chicago taxi mobility dataset using different approaches, where each pixel with the density value in the ranges[1] of $[d_{\min}, d_{\min} + 0.25(d_{\max} - d_{\min}))$, $[d_{\min} + 0.25(d_{\max} - d_{\min}), d_{\min} + 0.5(d_{\max} - d_{\min}))$, $[d_{\min} + 0.5(d_{\max} - d_{\min}), d_{\min} + 0.75(d_{\max} - d_{\min}))$, and $[d_{\min} + 0.75(d_{\max} - d_{\min}), d_{\max}]$ is colored by green, yellow, orange, and red, respectively. Here, $d_{\min}$ and $d_{\max}$ denote the minimum density value and the maximum density value of each figure, respectively.

[1] We use $[v_{\min}, v_{\max})$ to denote the range that includes $v_{\min}$ and excludes $v_{\max}$.

References

[1] ArcGIS. https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/how-line-density-works.htm.

[2] QGIS. https://docs.qgis.org/3.28/en/docs/user_manual/processing_algs/qgis/interpolation.html\#line-density.

[3] T. N. Chan, B. Zhu, D. Wu, Y. Peng, and L. H. U. LARGE: A length-aggregation-based grid structure for line density visualization. *Proc. VLDB Endow.*, 2024 (In submission).